



**SISTEM INFORMASI E-OFFICE PADA UIN MAHMUD YUNUS
BATUSANGKAR MENGGUNAKAN METODE MODEL VIEW
CONTROLLER (MVC)**

TUGAS AKHIR

Ditulis Sebagai Syarat untuk Memperoleh Gelar Ahli Madya (D.III)

Jurusan Manajemen Informatika

Diajukan Oleh:

DERRY DAVINCI

NIM: 1950401013

JURUSAN MANAJEMEN INFORMATIKA

FAKULTAS EKONOMI DAN BISNIS ISLAM

**UNIVERSITAS ISLAM NEGERI (UIN) MAHMUD YUNUS
BATUSANGKAR**

2022

ABSTRAK

Derry Davinci, Nim **1950401013**. Dengan judul **Sistem Informasi E-Office Pada UIN Mahmud Yunus Batusangkar Menggunakan Metode Model View Controller (MVC)**. Jurusan Manajemen Informatika, Fakultas Ekonomi dan Bisnis Islam, Universitas Islam Negeri (UIN) Mahmud Yunus Batusangkar.

E-Office merupakan pekerjaan yang berhubungan dengan administrasi perkantoran seperti pengelolaan surat masuk, surat keluar dan disposisi surat yang dikerjakan menggunakan bantuan alat komunikasi dan sistem informasi. Pada UIN Mahmud Yunus Batusangkar pengelolaan surat masuk masih dilakukan secara manual yaitu dengan melakukan pencatatan pada buku. Oleh karena itu dibutuhkan sebuah sistem informasi yang dapat mempermudah dalam pengelolaan administrasi persuratan pada UIN Mahmud Yunus Batusangkar.

Penelitian ini bertujuan untuk menghasilkan Sistem Informasi E-Office pada UIN Mahmud Yunus Batusangkar dengan menggunakan metode Model View Controller (MVC). Jenis penelitian ini adalah penelitian lapangan (Field Research) dimana penulis mendapatkan data langsung dari lapangan pada UIN Mahmud Yunus Batusangkar dan wawancara langsung dari pihak terkait, penelitian pustaka (Library Research) dengan membaca buku dan jurnal yang berkaitan dengan judul penelitian, dan penelitian labor (Laboratory Research) dengan menggunakan bantuan komputer sebagai alat bantu dalam pembuatan tugas akhir. Teknik pengumpulan data dalam penelitian ini adalah menggunakan wawancara dan pengumpulan dokumen-dokumen.

Hasil penelitian ini menunjukkan bahwa sistem yang dibangun dapat memberikan kemudahan dalam pelayanan surat masuk dan disposisi surat pada UIN Mahmud Yunus Batusangkar, sistem informasi *e-office* ini dapat meningkatkan kinerja dari petugas resepsionis dalam setiap layanan dan juga memberikan kemudahan bagi pimpinan dalam memberikan disposisi surat.

Kata Kunci : *Sistem Informasi, E-Office, , MVC*

DAFTAR ISI

ABSTRAK	i
DAFTAR ISI	ii
DAFTAR GAMBAR	iv
DAFTAR TABEL	v
BAB I PENDAHULUAN	1
A. Latar belakang	1
B. Identifikasi Masalah	3
C. Batasan Masalah	4
D. Rumusan Masalah	4
E. Tujuan Penelitian	4
F. Manfaat Penelitian	4
G. Metodologi Penelitian	5
H. Sistematika Penulisan	6
BAB II LANDASAN TEORI	7
A. Gambaran Umum	7
1. Sejarah UIN Mahmud Yunus Batusangkar	7
2. Visi dan Misi UIN Mahmud Yunus Batusangkar	10
3. Tujuan	11
4. Tugas Pokok dan Fungsi	12
5. Struktur Organisasi	14
B. Konsep Dasar Sistem Informasi	15
1. Sistem	15
2. Informasi	19
3. Sistem Informasi	21
C. E-Office	23
1. Definisi Electronic Office (E-Office)	23
2. Surat Masuk	24
3. Disposisi	24

D. Model View Controller (MVC)	25
E. Alat Bantu Pengembangan Sistem.....	28
1. Unified Modeling Language (UML)	28
2. Entity Relationship Diagram (ERD).....	33
F. Perangkat Lunak Pembangun Sistem.....	34
1. PHP	34
2. <i>PhpMyAdmin</i>	36
3. <i>Mysql</i>	37
4. XAMPP	40
5. <i>Framework Codeigniter</i>	40
6. <i>Visual Studio Code</i>	43
BAB III ANALISA DAN HASIL	44
A. Analis Sistem	44
1. Analisis Sistem Yang Sedang Berjalan	44
2. Evaluasi Sistem Yang Sedang Berjalan.....	45
3. Penerapan Konsep MVC pada <i>E-Office</i>	45
B. Perancangan Sistem	47
1. Perancangan Global	47
2. Perancangan Terinci	58
BAB IV PENUTUP.....	67
A. Kesimpulan.....	67
B. Saran.....	67
DAFTAR PUSTAKA	68

DAFTAR GAMBAR

Gambar 2. 1 Struktur Organisasi UIN Mahmud Yunus	14
Gambar 2. 2 Daur Hidup Sistem (<i>System Life Cycle</i>)	17
Gambar 2. 3 Siklus Informasi	20
Gambar 2. 4 Alur Kerja MVC	27
Gambar 2. 5 Alur kerja <i>CodeIgniter</i>	41
Gambar 3. 1 Konsep MVC pada E-Office	46
Gambar 3. 2 <i>Use Case Diagram</i>	48
Gambar 3. 3 <i>Class Diagram</i>	50
Gambar 3. 4 <i>Activity Diagram</i> Resepsionis.....	51
Gambar 3. 5 <i>Activity Diagram</i> Pimpinan	52
Gambar 3. 6 <i>Sequence Diagram</i> MVC	53
Gambar 3. 7 <i>Sequence Diagram</i> Resepsionis	53
Gambar 3. 8 <i>Sequence Diagram</i> Pimpinan	54
Gambar 3. 9 <i>Collaboration Diagram</i> Kelola Surat Masuk Resepsionis.....	54
Gambar 3. 10 <i>Collaboration Diagram</i> Disposisi Pimpinan	55
Gambar 3. 11 <i>Collaboration Diagram</i> Tracking Disposisi.....	55
Gambar 3. 12 <i>Collaboration Diagram</i> Pengarsipan Surat Pimpinan	56
Gambar 3. 13 <i>Collaboration Diagram</i> Unduh Laporan.....	56
Gambar 3. 14 <i>Entity Relationship Diagram</i> (ERD).....	57
Gambar 3. 15 Laporan Disposisi	59
Gambar 3. 16 Laporan Surat Masuk	60
Gambar 3. 17 Tampilan Tracking Disposisi Surat.....	60
Gambar 3. 18 Tampilan <i>Login</i>	61
Gambar 3. 19 Tampilan <i>Input</i> Surat Masuk	62
Gambar 3. 20 Tampilan <i>Input</i> Disposisi	63

DAFTAR TABEL

Tabel 2. 1 Pimpinan UIN Mahmud Yunus Batusangkar (IAIN Batusangkar, 2022)	10
Tabel 2. 2 <i>Use Case Diagram</i>	29
Tabel 2. 3 <i>Use Case Diagram</i>	30
Tabel 2. 4 <i>Use Case Diagram</i>	31
Tabel 2. 5 <i>Sequence Diagram dan Collaboration Diagram</i>	32
Tabel 2. 6 Entity Relationship Diagram.....	33
Tabel 3. 1 User.....	63
Tabel 3. 2 Unit	64
Tabel 3. 3 Jabatan	64
Tabel 3. 4 Surat Masuk	64
Tabel 3. 5 Disposisi.....	65

BAB I

PENDAHULUAN

A. Latar belakang

Pada saat ini perkembangan teknologi begitu pesat khususnya dibidang informasi. Sistem informasi adalah salah satu dari perkembangan tersebut, dengan adanya sistem informasi, pengelolaan informasi menjadi jauh lebih efisien. Meskipun begitu, dalam beberapa kasus sistem informasi belum diterapkan secara maksimal seperti dalam pengelolaan administrasi. Masih adanya penggunaan buku dan kertas dalam mendata administrasi seperti persuratan yang dilakukan secara manual, membuat administrasi tersebut menjadi tidak efisien dan banyak memakan waktu, sehingga memengaruhi kinerja manajemen informasi (Syafitri, 2020). Dalam perkantoran modern ini segala kegiatan manajemen perkantoran dikerjakan dengan menggunakan alat-alat otomatisasi perkantoran, diantaranya adalah perangkat komputer dengan berbagai aplikasi salah satunya yaitu aplikasi perkantoran elektronik atau *E-Office* (Julianty et al., 2019).

Electronic Office (e-office) adalah suatu sistem yang berhubungan dengan administrasi secara maya memusatkan komponen-komponen sebuah organisasi dimana data, informasi dan komunikasi dibuat melalui media telekomunikasi (Ardhianto, 2015). Jadi dapat diartikan bahwa *e-office* adalah pekerjaan yang berhubungan dengan administrasi perkantoran seperti pengelolaan surat masuk, surat keluar dan disposisi surat yang dikerjakan menggunakan bantuan alat komunikasi dan sistem informasi. Dengan adanya sistem informasi *e-office* akan dapat meningkatkan kinerja perkantoran, khususnya dalam pengolahan administrasi surat.

Sistem informasi *e-office* yang akan dibangun menerapkan metode *Model View Controller* (MVC). *Model View Controller* (MVC) merupakan model pembuatan program dengan arsitektur yang memisahkan proses, tampilan serta penghubung proses dan tampilan (Badiyanto, 2013). Tujuan dari penerapan MVC adalah untuk menyederhanakan *script* dan memudahkan

programmer dalam memelihara sistem, karena data (*model*) terpisah dari tampilan (*view*) juga pengolahannya (*controller*), sehingga *programmer* tidak kesulitan saat memperbaiki bug pada sistem, atau bahkan melakukan pengembangan sistem lanjutan (Sudradjat et al., 2022). Jadi dapat disimpulkan *Model View Controller* (MVC) adalah sebuah metode pemrograman yang dipisah menjadi tiga bagian, yaitu *model* yang akan melakukan pemrosesan data, *view* yang akan menampilkan data yang diproses oleh *model* dan *controller* sebagai penghubung antara *model* dan *view* sehingga kodingan menjadi lebih sederhana dan memudahkan dalam pengembangan sistem lanjutan.

Beberapa penelitian sebelumnya yang melandasi penelitian ini adalah penelitian yang dilakukan oleh (Julianty et al., 2019), permasalahan yang dihadapi dalam penelitian ini yaitu pengelolaan data surat menyurat yang masih dilakukan dengan cara manual dengan menggunakan buku agenda. Penelitian ini menghasilkan aplikasi *E-Office* berbasis web yang dapat mempermudah penyimpanan, pengelolaan serta pencarian kembali data surat karena data disimpan menggunakan *database*. Selanjutnya penelitian yang dilakukan oleh (Syafitri et al., 2020), permasalahan yang dihadapi juga sama yaitu proses belum adanya sistem informasi administrasi surat sehingga pengelolaan surat masih dilakukan secara manual dengan menggunakan kertas. Penelitian ini menghasilkan sistem informasi administrasi persuratan (*Paperless office*) yang mengurangi penggunaan kertas dalam administrasi surat serta penyimpanan dan pengelolaan surat menjadi lebih mudah dan cepat.

Berdasarkan pengamatan dan wawancara yang telah dilakukan pada Universitas Islam Negeri Mahmud Yunus Batusangkar saat ini proses pengelolaan surat masuk dan disposisi surat masih dilakukan secara manual yaitu dengan menggunakan lembar disposisi. Proses disposisi surat seperti ini seringkali mengalami penundaan jika pimpinan sedang berada di luar kota atau tidak berada di kantor. Keberadaan posisi surat juga tidak dapat pantau secara *realtime* karena keberadaan posisi surat hanya dapat diketahui setelah

surat dilakukan disposisi oleh pimpinan. Membutuhkan waktu yang lama untuk pencarian kembali data-data surat masuk yang telah berlalu. Permasalahan tersebut dapat menghambat proses disposisi surat dan bisa menjadi salah satu faktor penghambat bagi aktivitas operasional lainnya yang ada di UIN Mahmud Yunus Batusangkar.

Berdasarkan permasalahan di atas, diperlukan suatu sistem informasi yang dapat memudahkan dalam proses pengelolaan surat masuk dan disposisi surat yang ada pada UIN Mahmud Yunus Batusangkar. Penggunaan metode *Model View Controller* (MVC) dalam sistem *e-office* ini bertujuan untuk memudahkan dalam membuat koding pemrograman menjadi lebih sederhana dan rapi serta memudahkan dalam mengatasi *bug* dan pengembangan sistem lanjutan oleh *programmer* lain. Oleh karena itu, penulis tertarik melakukan penelitian dan merancang sistem informasi dengan judul **“SISTEM INFORMASI E-OFFICE PADA UIN MAHMUD YUNUS BATUSANGKAR MENGGUNAKAN METODE MODEL VIEW CONTROLLER (MVC)”**.

B. Identifikasi Masalah

Berdasarkan latar belakang yang ada dapat diidentifikasi beberapa permasalahan yang muncul antara lain :

1. Proses disposisi surat masuk yang masih dilakukan secara manual yaitu dengan menggunakan media kertas berupa lembar disposisi.
2. Tertundanya proses disposisi surat jika pimpinan sedang berada di luar kota atau tidak berada di kantor sehingga waktu yang dibutuhkan dalam memproses surat yang masuk menjadi lama.
3. Keberadaan posisi disposisi surat yang tidak dapat dipantau secara *realtime*.
4. Pencarian kembali data-data surat membutuhkan waktu yang lama karena harus dicari satu persatu.

C. Batasan Masalah

Berdasarkan identifikasi masalah dan mengingat keterbatasan waktu dan biaya, maka penulis membatasi permasalahan hanya membahas sistem informasi e-office mengenai surat masuk dan disposisi surat pada Rektorat UIN Mahmud Yunus Batusangkar menggunakan Metode *Model View Controller* (MVC).

D. Rumusan Masalah

Berdasarkan latar belakang masalah dan identifikasi masalah, maka dapat dirumuskan masalah penelitian yaitu :

1. Apakah sistem informasi *e-office* menggunakan metode MVC mampu memberikan solusi terhadap disposisi surat masuk pada UIN Mahmud Yunus Batusangkar?
2. Bagaimana cara merancang sistem informasi *e-office* pada UIN Mahmud Yunus Batusangkar menggunakan metode MVC?

E. Tujuan Penelitian

Dalam penulisan laporan ini memiliki tujuan yang ingin dicapai, adapun tujuan tersebut adalah :

1. Untuk menghasilkan sistem informasi *e-office* dengan *database* dan kode pemrograman yang serdehana, rapi dan baik sehingga proses pengelolaan surat menyurat, pelaporan dan pencarian kembali data pelayanan menjadi lebih cepat serta proses pengembangan aplikasi menjadi lebih mudah.
2. Untuk merancang sistem informasi *e-office* menggunakan metode MVC supaya nantinya bisa menjadi alat bantu dalam proses pengelolaan surat masuk dan disposisi pimpinan.

F. Manfaat Penelitian

Hasil penelitian ini berguna untuk :

- a. Bagi Mahasiswa/Dosen/Masyarakat luar

- 1) Mendapatkan balasan surat dengan cepat karena adanya sistem informasi e-office.
 - 2) Data pengguna juga aman tersimpan dengan adanya database yang baik.
- b. Bagi Lembaga
- 1) Untuk mempermudah pekerjaan staf dan semua pegawai UIN Mahmud Yunus Batusangkar.
 - 2) Dalam menerima surat masuk akan lebih efisien dan lebih cepat dengan adanya sistem informasi *e-office*.
 - 3) Dalam pembuatan laporan seluruh atau data tertentu dari data surat masuk akan lebih mudah karena adanya *database* yang baik.

G. Metodologi Penelitian

Adapun metodologi yang digunakan dalam penelitian ini adalah sebagai berikut :

a. *Field Research* (Studi Lapangan)

Data yang diperoleh dalam penelitian di lapangan ini untuk melihat kenyataan yang sebenarnya dalam masalah yang diteliti dengan cara sebagai berikut :

1) *Observasi*

Teknik pengumpulan data dengan mengadakan penelitian dan peninjauan langsung ke staf pimpinan UIN Mahmud Yunus Batusangkar.

2) *Interview*

Dalam penelitian ini, wawancara dilakukan dengan tujuan untuk mendefinisikan dan menganalisa kebutuhan perangkat lunak yang akan dikembangkan.

b. *Library Research* (Studi Pustaka)

Penelitian kepustakaan ini ialah pengumpulan data dengan cara mempelajari berbagai literatur, buku, hasil penelitian yang sejenis dan media lain yang mempunyai kaitan dengan masalah dan tema penelitian.

c. *Laboratory Research* (Penelitian Labor)

Penelitian ini dilakukan untuk mempraktek secara langsung analisa serta pembuatan program yang berguna untuk pengembangan sistem baru yang bermaksud dengan menggunakan komputer sebagai alat bantu dalam penyelesaian tugas akhir ini.

H. Sistematika Penulisan

Sistematika penulisan dapat diartikan sebagai cara yang digunakan dalam pembuatan laporan untuk memberikan gambaran isi tugas akhir yang terdiri pendahuluan, landasan teori, analisa dan hasil serta penutup.

Sistematika penulisan laporan tugas akhir ini dibagi dalam IV bab yang disusun sebagai berikut :

BAB I Pendahuluan, Bab ini merupakan penguraian mengenai latar belakang masalah, identifikasi masalah, batasan masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

BAB II Landasan Teori, Bab ini berisi teori-teori dasar mengenai Sistem Informasi *E-Office* Pada UIN Mahmud Yunus Batusangkar Menggunakan Metode *Model View Controller* (MVC).

BAB III Analisa Dan Hasil, Bab ini membahas analisa dan hasil perancangan Sistem Informasi *E-Office* Pada UIN Mahmud Yunus Batusangkar Menggunakan Metode *Model View Controller* (MVC).

BAB IV Penutup, Bab ini berisi kesimpulan yang didapat selama pembuatan laporan tugas akhir serta saran-saran yang akan menjadi masukan bagi perkembangan sistem selanjutnya.

BAB II

LANDASAN TEORI

A. Gambaran Umum

1. Sejarah UIN Mahmud Yunus Batusangkar

Upaya pembentukan sebuah Perguruan Tinggi Agama Islam di Batusangkar yang akhirnya berdiri secara resmi pada tahun 1968, dimulai dari pembentukan Panitia Persiapan Pendirian Perguruan Tinggi Agama Islam yang diketahui oleh Mahyudin Algamar (Bupati Tanah Datar pada waktu itu). Dengan melihat personil yang berperan semakin menampakkan adanya indikator bahwa pendirian Perguruan Tinggi Agama Islam di Batusangkar memang didukung oleh semua unsur, baik pemerintah maupun masyarakat.

Setelah Panitia Persiapan Perguruan Tinggi Agama Islam berjalan beberapa bulan, statusnya ditingkatkan menjadi Fakultas Tarbiyah Swasta yang berlokasi di Kubu Rajo Lima Kaum Batusangkar di atas tanah seluas 11.026 M2. Usaha ini semakin terdukung oleh tingginya keantusiasan masyarakat Tanah Datar. Hal ini dibuktikan dengan adanya kegairahan putra-putri mereka untuk melanjutkan pendidikan di Fakultas ini. Realitas tersebut menjadi potensi utama untuk beralih status dari swasta menjadi sebuah Fakultas yang berada dalam naungan Institut Agama Islam Negeri (IAIN) Imam Bonjol Padang. Peralihan tersebut berdasarkan Keputusan Menteri Agama RI No. 238 tanggal 20 Mei 1971. Dengan demikian Fakultas swasta ini resmi menjadi Fakultas Tarbiyah Negeri dengan status Fakultas Muda, artinya hanya bisa membuka program pendidikan tingkat Sarjana Muda.

Di tengah perjalanannya, pada tahun 1974 Fakultas Tarbiyah IAIN Imam Bonjol Batusangkar diguncang oleh suatu peraturan rasionalisasi fakultas dalam lingkungan IAIN se-Indonesia. Sehingga Fakultas Tarbiyah di Batusangkar tidak dibenarkan lagi menerima mahasiswa baru karena

akan ditarik ke IAIN Imam Bonjol Padang. Namun, berkat usaha sungguh-sungguh dari civitas akadenika Fakultas Tarbiyah IAIN Imam Bonjol Batusangkar dan masyarakat serta Pemerintah Daerah Kabupaten Tanah Datar untuk mempertahankannya, maka Fakultas Tarbiyah IAIN Imam Bonjol Batusangkar ini tidak jadi ditarik ke Padang. Untuk itu pada tahun 1976, Rektor IAIN Imam Bonjol Padang Drs. Sanusi Latief memperkenalkan Fakultas Tarbiyah IAIN Imam Bonjol Batusangkar menerima mahasiswa baru dengan beberapa persyaratan: pertama, harus ada asrama mahasiswa. Kedua, harus ada Yayasan Penyantun. Ketiga, jumlah mahasiswa baru minimal 40 orang. Semua persyaratan tersebut akhirnya dapat dipenuhi oleh civitas akademiknya atas dukungan dan perhatian tokoh-tokoh eksternal.

Selanjutnya, berdasarkan Keputusan Menteri Agama No. 69/1982 Fakultas Tarbiyah ini meningkat statusnya dari Fakultas Muda menjadi Fakultas Madya. Dengan demikian semenjak tahun 1982, Fakultas Tarbiyah IAIN Imam Bonjol Batusangkar berhak menyelenggarakan perkuliahan tingkat doktoral dengan Jurusan Pendidikan Agama Islam (PAI).

Pada tahun akademik 1992/1993, Fakultas Tarbiyah membuka Jurusan Pendidikan Bahasa Arab (PBA). Pembukaan Jurusan baru ini disebabkan adanya tuntutan bahwa setiap Fakultas Madya diwajibkan memiliki sekurang-kurangnya dua Jurusan. Pada tahun akademik 1996/1997, Fakultas Tarbiyah IAIN Imam Bonjol Batusangkar membuka lagi Jurusan Kependidikan Islam (KI).

Setelah kurang lebih 26 tahun Fakultas Tarbiyah IAIN Imam Bonjol Batusangkar berada dalam lingkungan IAIN Imam Bonjol Padang, maka pada tahun 1997 berubah statusnya menjadi Sekolah Tinggi Agama Islam Negeri (STAIN) Batusangkar. Perubahan status ini berdasarkan Kepres No. 11/1997 dan Surat Keputusan Menteri Agama RI No. 285/1997, dan memberi akses kepada STAIN untuk “duduk sama rendah dan berdiri sama tinggi” dengan berbagai perguruan tinggi lainnya.

Dengan adanya perubahan status ini, maka seluruh dosen dan karyawan beserta sarana dan prasarana Fakultas Tarbiyah IAIN Imam Bonjol Batusangkar beralih menjadi asset STAIN Batusangkar.

STAIN Batusangkar pada saat perubahan status memiliki dua jurusan yaitu Jurusan Tarbiyah dan Jurusan Syariah. Jurusan Tarbiyah pada waktu itu terdiri atas tiga Program Studi: Pendidikan Agama Islam (PAI), Pendidikan Bahasa Arab (PBA) dan Kependidikan Islam (KI). Sementara itu, Jurusan Syariah pada awal berdirinya tahun hanya mempunyai satu program studi yaitu Program Studi Ahwal al-Syakhshiyah.

Perkembangan program studi tahap selanjutnya adalah pada tahun 2000 dengan keluarnya izin penyelenggaraan Program Studi Muamalah (Ekonomi Islam Konsentrasi Perbankan Syariah) dan Tadris Bahasa Inggris. Selanjutnya pada tahun 2005 dibuka Program Studi Hukum Ekonomi Syariah, tahun 2011 dibuka program studi Ekonomi Syariah pada Jurusan Syariah, kemudian pada tahun 2012 dibuka Program Studi Bimbingan Konseling, Tadris Fisika dan Tadris Biologi pada Jurusan Tarbiyah serta Program Studi Manajemen Informatika (D3). Pengalaman STAIN dalam mengelola pendidikan pada jenjang Strata 1 dan Diploma III, juga telah menantang STAIN Batusangkar untuk menyelenggarakan jenjang pendidikan yang lebih tinggi yaitu untuk Program Magister. Hal ini terwujud dengan keluarnya izin penyelenggaraan Program Studi S2 Manajemen Pendidikan Islam pada tahun 2010 dan Hukum Ekonomi Syariah tahun 2012.

Pada tahun 2014 STAIN Batusangkar kembali diberi kepercayaan oleh Kementrian Agama untuk menyelenggarakan 6 (enam) program studi baru. Keenam program studi baru tersebut adalah Pendidikan Guru Raudhatul Athfal (PGRA), Hukum Tata Negara Islam (Siyasah), Ilmu Quran Tafsir, Ilmu Hadis, Komunikasi Penyiaran Islam, dan Program Studi S Pendidikan Agama Islam.

Setelah melalui proses yang cukup panjang, maka pada akhir tahun 2015 Sekolah Tinggi Agama Islam Negeri (STAIN) Batusangkar resmi menjadi Institut Agama Islam Negeri (IAIN) Batusangkar melalui Perpres No. 147 tahun 2015, tanggal 23 Desember 2015.

Setelah melalui proses yang cukup Panjang, maka pertengahan tahun 2022 Institut Agama Islam Negeri (IAIN) Batusangkar, resmi menjadi Universitas Islam Negeri (UIN) Mahmud Yunus Batusangkar melalui keputusan presiden Nomor 84 tahun 2022.

Sepanjang sejarahnya, tokoh-tokoh yang pernah memimpin perguruan tinggi ini dari Fakultas Tarbiyah IAIN Imam Bonjol di Batusangkar sampai menjadi IAIN Batusangkar adalah:

Tabel 2. 1 Pimpinan UIN Mahmud Yunus Batusangkar
(IAIN Batusangkar, 2022)

No	Nama	Tahun
1.	Drs. H. Haitami	1971-1977
2.	Drs. Thamsir Thain Burhani	1977-1985
3.	Drs. H. Haitami	1985-1989
4.	Drs. Fachri Syamsudin	1989-1992
5.	Drs. Arpinus	1992-1995
6.	Prof. Dr. H. Ramayulis	1996-2002
7.	Drs. H. Syukri Iska, M.Ag.	2002-2010
8.	Prof. Dr. H. Hasan Zaini, MA.	2010-2014
9.	Dr. Kasmuri, MA.	2014-2020
10.	Prof. Dr. Marjoni Imamora M.Sc	2020-sekarang

2. Visi dan Misi UIN Mahmud Yunus Batusangkar

a. Visi

“Menjadi Lembaga Pendidikan Tinggi Islam berkelas internasional dengan penguatan keilmuan yang *integrative dan inter-konektif*, berbasis riset dan kearifan lokal”.

b. Misi

- 1) Menyelenggarakan pendidikan tinggi Islam yang berdaya saing Internasional untuk kepentingan umat, bangsa dan kemanusiaan.
- 2) Melaksanakan pendidikan atau pengajaran secara integrative dan interkonektif yang relevan dengan perkembangan keilmuan internasional dan tuntutan pengguna.
- 3) Melaksanakan penelitian integrative, interkonektif dan berbasis kearifan lokal.
- 4) Melaksanakan pengabdian untuk kesejahteraan masyarakat yang berbasis riset dan kearifan lokal.
- 5) Menciptakan lulusan yang cerdas secara intelektual, spiritual, emosional, sosial dan berdaya saing dalam dunia kerja.
- 6) Mengelola pendidikan tinggi secara professional, modern dan akuntabel.

3. Tujuan

- a. Terwujudnya Program Studi yang unggul dalam pengembangan keilmuan yang interaktif dan inter-konektif.
- b. Terbangunnya iklim akademik yang mendukung terhadap pelaksanaan Tri Dharma Perguruan tinggi berbasis riset dan kearifan lokal.
- c. Terwujudnya hasil riset yang kompetitif dan berdaya guna untuk umat, bangsa dan kemanusiaan.
- d. Penguatan sumber daya pendidik dan tenaga kependidikan yang cerdas dan profesional.
- e. Terwujudnya lulusan yang cerdas secara intelektual, spiritual, emosional, sosial, dan berdaya saing dalam dunia kerja.

- f. Terbangunnya tata kelola yang akuntabel, bersih dan modern berbasis ICT (*Information, Communication and Technology*).
- g. Bertambahnya kerjasama dengan berbagai pihak dalam pencapaian visi dan misi institusi.

4. Tugas Pokok dan Fungsi

a. Tugas Pokok

Adapun tugas pokok UIN Mahmud Yunus Batusangkar adalah sebagai berikut:

- 1) Menyelenggara program pendidikan akademik atau profesi dalam bidang ilmu keIslaman dan ilmu lain yang terkait, dalam rangka menghasilkan lulusan yang berkualitas, berdaya saing tinggi, dan bermanfaat bagi masyarakat.
- 2) Melakukan penelitian dalam bidang ilmu ke Islaman dan ilmu lain yang terkait, dalam rangka menghasilkan hasil penelitian yang berkualitas dan bermanfaat bagi pengembangan ilmu pengetahuan dan/atau pemcahan masalah d masyarakat, dan
- 3) Melaksanakan kegiatan pengabdian kepada masyarakat di sebuah rangka menyumbangkan manfaat hasil pendidikan dan penelitian.

b. Fungsi

Untuk melaksanakan tugas pokok tersebut Universitas Islam Negeri Batusangkar sebagai berikut:

- 1) Perumusan kebijakan dalam perencanaan program,
- 2) Pelaksanaan pendidikan dan pengajaran, penelitian, pengembangan ilmu pengetahuan agama Islam, seni, dan pengabdian pada masyarakat,
- 3) Pembinaan civitas akademika dan hubungan akademik ilmiah dan sosial sesuai dengan lingkungannya,

- 4) Pelaksanaan kerjasama Sekolah Tinggi dengan perguruan tinggi dan/atau lembaga-lembaga lain dalam dan luar negeri, dan
- 5) Pelaksanaan kegiatan pelayanan administratif.

5. Struktur Organisasi



Gambar 2. 1 Struktur Organisasi UIN Mahmud Yunus

B. Konsep Dasar Sistem Informasi

1. Sistem

a. Pengertian Sistem

Menurut (Abdul, 2014), Sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Menurut (Taufiq, 2003), Sistem adalah kumpulan dari sub-sub sistem yang saling berhubungan dan bekerja sama. Sedangkan menurut (Sutabri, 2012), Sistem adalah sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu.

Berdasarkan definisi diatas, dapat disimpulkan pengertian sistem adalah sekumpulan elemen dari sub-sub sistem atau komponen atau unsur-unsur yang terpadu, saling terkait atau saling berinteraksi dan saling berhubungan satu sama lain untuk mencapai suatu tujuan.

b. Daur Hidup Sistem

Menurut (Sutabri, 2012), siklus hidup sistem (*system life cycle*) adalah proses *evolusioner* yang diikuti dalam menerapkan sistem atau subsistem informasi berbasis komputer. Fase atau tahapan dari daur hidup suatu sistem :

1) Mengenali adanya kebutuhan

Sebelum segala sesuatunya terjadi, timbul suatu kebutuhan yang harus dapat dikenali. Kebutuhan dapat terjadi sebagai hasil pengembangan dari organisasi dan volume yang meningkat melebihi kapasitas dari sistem yang ada. Suatu kebutuhan ini harus dapat didefinisikan dengan jelas. Tanpa adanya kejelasan dari kebutuhan yang ada, pembangunan sistem akan kehilangan arah dan efektifitasnya.

2) Pembangunan sistem

Suatu proses atau perangkat prosedur yang harus diikuti untuk menganalisa kebutuhan yang timbul dan membangun suatu sistem untuk dapat memenuhi kebutuhan tersebut.

3) Pemasangan sistem

Setelah tahap pembangunan sistem selesai, sistem akan dioperasikan. Pemasangan sistem merupakan tahap yang penting dalam daur hidup sistem. Didalam peralihan dari tahap pembangunan menuju tahap operasional terjadi pemasangan sistem yang sebenarnya yang merupakan langkah akhir dari suatu pembangunan sistem.

4) Pengoperasian sistem

Program-program computer dan prosedur-prosedur pengoperasian yang membentuk suatu sistem informasi semuanya bersifat statis, sedangkan organisasi ditunjang oleh sistem informasi tadi. Ia selalu mengalami perubahan-perubahan itu karena pertumbuhan kegiatan bisnis, perubahan peraturan, dan kebijaksanaan ataupun kemajuan teknologi. Untuk perubahan-perubahan tersebut, sistem harus diperbaiki atau diperbaharui.

5) Sistem menjadi usang

Kadang perubahan yang terjadi begitu drastis sehingga tidak dapat diatasi hanya dengan melakukan perbaikan-perbaikan pada sistem yang berjalan. Tibalah saatnya secara ekonomis dan teknik sistem yang ada sudah tidak layak lagi untuk dioperasikan dan sistem yang baru perlu dibangun untuk menggantikannya.

Berikut adalah contoh gambar daur hidup sistem menurut (Sutabri, 2012), sebagai berikut :



Gambar 2. 2 Daur Hidup Sistem (*System Life Cycle*)

c. Karakteristik Sistem

Menurut (Hartono, 2013), bahwa sebuah sistem memiliki paling sedikit sepuluh karakteristik sebagai berikut :

1) Komponen (*Components*)

Bagian-bagian atau elemen-elemen yang dapat berupa benda atau manusia, berbentuk nyata atau abstrak, dan disebut subsistem.

2) Penghubung antar bagian (*Interface*)

Sesuatu yang bertugas menjembatani satu bagian dengan bagian lain, dan memungkinkan terjadinya interaksi/komunikasi antar bagian.

3) Batas (*Boundary*)

Sesuatu yang membedakan antara satu sistem dengan sistem atau dengan sistem-sistem lain.

4) Lingkungan (*Environment*)

Segala sesuatu yang berada diluar sistem dan dapat bersifat menguntungkan atau merugikan sistem yang bersangkutan.

5) Masukan (*Input*)

Sesuatu yang merupakan bahan untuk diolah atau diproses oleh sistem.

6) Mekanisme Pengolahan (*Processing*)

Perangkat dan prosedur untuk mengubah masukan menjadi keluaran dan menampilkannya.

7) Keluaran (*Output*)

Berbagai macam bentuk hasil atau produk yang dikeluarkan dari pengolahan.

8) Tujuan (*Goal/Objective*)

Sesuatu atau keadaan yang ingin dicapai oleh sistem, baik dalam jangka pendek maupun jangka panjang.

9) Sensor dan Kendali (*Sensor and Control*)

Sesuatu yang bertugas untuk memantau dan menginformasikan perubahan-perubahan didalam lingkungan dan dalam diri sistem kepada sistem.

10) Umpan balik (*Feedback*)

Informasi tentang perubahan-perubahan lingkungan dan perubahan-perubahan (penyimpangan) dalam diri sistem.

d. Klarifikasi Sistem

Menurut (Yakub, 2012), Sistem dapat diklasifikasikan dari beberapa sudut pandang diantaranya :

1) Sistem abstrak (*abstract system*)

Sistem Abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Sistem teologia yang berisi gagasan tentang hubungan manusia dengan Tuhan merupakan contoh *abstract system*.

2) Sistem fisik (*physical system*)

Sistem fisik adalah sistem yang ada secara fisik, Sistem komputer, sistem akuntansi, sistem produksi, sistem sekolah, dan sistem transportasi merupakan contoh *physical system*.

3) Sistem tertentu (*deterministic system*)

Sistem tertentu adalah sistem yang beroperasi dengan tingkah laku yang dapat diprediksi, interaksi antara bagian dapat dideteksi dengan pasti sehingga keluarannya dapat diramalkan. Sistem

komputer sudah diprogramkan, merupakan contoh *deterministic system* karena program komputer dapat diprediksi dengan pasti.

4) Sistem tak tentu (*probabilistic system*)

Sistem tak tentu adalah suatu sistem yang kondisi masa depannya tidak dapat diprediksikan karena mengandung unsur probabilitas. Sistem arisan merupakan contoh *probabilistic system* karena sistem arisan tidak dapat diprediksikan dengan pasti.

5) Sistem tertutup (*close system*)

Sistem tertutup merupakan sistem yang tidak bertukar materi, informasi, atau energi dengan lingkungan. Sistem ini tidak berinteraksi dan tidak dipengaruhi oleh lingkungan, misalnya reaksi kimia dalam tabung terisolasi.

6) Sistem terbuka (*open system*)

Sistem ini adalah sistem yang berhubungan dengan lingkungan dan dipengaruhi oleh lingkungan. Sistem perdagangan merupakan contoh *open system*, karena dapat dipengaruhi oleh lingkungan.

2. Informasi

a. Pengertian Informasi

Untuk menganalisa suatu sistem harus mengerti terlebih dahulu komponen-komponen yang ada dalam sistem tersebut. Dari mana data dan informasi tersebut diperoleh dan kemana hasil pengolahan data dan informasi tersebut diperlukan definisi informasi terdapat beberapa pandangan, diantaranya yaitu :

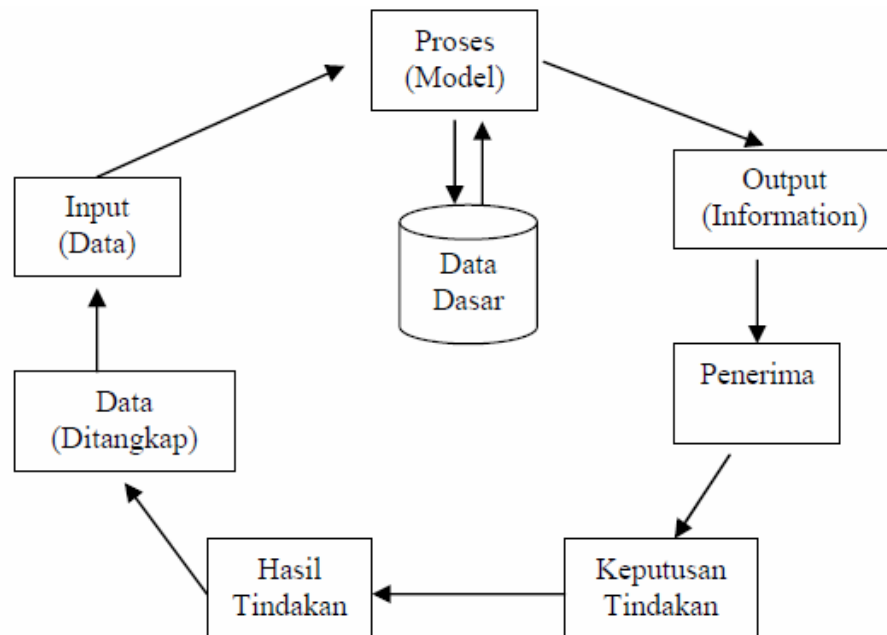
- 1) Menurut (Taufiq, 2003), Informasi adalah data-data yang diolah sehingga memiliki nilai tambah dan bermanfaat bagi pengguna.
- 2) Menurut (Maimunah et al., 2012), Informasi adalah data yang telah diolah menjadi sebuah bentuk yang lebih berarti bagi penerimanya dan bermanfaat dalam mengambil suatu keputusan.

- 3) Menurut (Sutabri, 2012), Informasi adalah data yang telah diklasifikasi atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan.

Dari beberapa definisi diatas, maka dapat disimpulkan bahwa informasi adalah data yang telah diolah atau diklasifikasi menjadi sesuatu yang memiliki arti dan manfaat bagi pengguna atau penerimanya dalam proses pengambilan keputusan.

b. Siklus Informasi

Pengolahan data menjadi suatu informasi dapat digambarkan sebagai sebuah siklus yang berkaitan antara satu dengan yang lain seperti gambar dibawah ini.



Gambar 2. 3 Siklus Informasi

Siklus informasi adalah gambaran secara keseluruhan mengenai proses terhadap data sehingga menjadi informasi yang berguna bagi pengguna. Proses menghasilkan informasi harus melalui tahapan-tahapan yang dilakukan komputer sebagai teknologi informasi. Tahapan-tahapan tersebut terdiri dari input, proses, output yang disebut sebagai siklus proses informasi.

Penjelasan dari Gambar, data diolah melalui suatu model menjadi sebuah informasi, penerima kemudian menerima informasi

tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan tindakan lainnya. Data yang di tangkap dianggap sebagai input di proses kembali melalui model, dan begitu seterusnya dan membentuk sebuah siklus yang dinamakan siklus informasi.

c. Nilai Informasi

Menurut (Sutabri, 2012), nilai informasi ditentukan oleh 2 (dua) hal, yaitu manfaat dan biaya untuk mendapatkannya. Suatu informasi dikatakan bernilai bila manfaat lebih efektif dibanding dengan biaya mendapatkannya.

d. Kualitas Informasi

Menurut (Sutabri, 2012), kualitas suatu informasi tergantung 3 (tiga) hal, yaitu :

1) Akurat (*accurate*)

Informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan.

2) Tepat waktu (*timeline*)

Informasi yang datang pada si penerima tidak boleh terlambat.

3) Relevan (*relevance*)

Informasi tersebut mempunyai manfaat untuk pemakaiannya.

Berdasarkan pengertian kualitas informasi diatas, dapat disimpulkan bahwa informasi yang di dapat harus akurat, tepat waktu serta relevan dalam memberikan informasi yang berkualitas dan memiliki nilai ekonomi juga dapat dipercaya.

3. Sistem Informasi

a. Pengertian Sistem Informasi

Menurut (Taufiq, 2003), Sistem informasi adalah kumpulan dari sub-sub sistem yang saling terintegrasi dan berkolaborasi untuk menyelesaikan masalah tertentu dengan cara mengolah data dengan alat yang namanya komputer sehingga memiliki nilai tambah dan bermanfaat bagi pengguna. Menurut (Sutabri, 2012), Sistem informasi

adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan.

Berdasarkan kedua definisi diatas, dapat disimpulkan bahwa sistem informasi adalah suatu kumpulan dari sub-sub sistem yang saling berhubungan yang diolah terlebih dahulu sehingga bermanfaat bagi organisasi.

b. **Komponen Sistem Informasi**

Menurut (Abdul, 2014), sistem informasi mengandung komponen-komponen seperti berikut :

- 1) Perangkat keras (*hardware*), yang mencakup piranti-piranti fisik seperti komputer dan printer.
- 2) Perangkat lunak (*software*) atau program, yaitu sekumpulan instruksi yang memungkinkan perangkat keras memproses data.
- 3) Prosedur, yaitu sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dan pembangkitan keluaran yang dikehendaki.
- 4) Orang, yakni semua pihak yang bertanggung jawab dalam pengembangan sistem informasi, pemrosesan, dan penggunaan keluaran sistem informasi.
- 5) Basis data (*database*), yaitu kumpulan tabel, hubungan, dan lain-lain yang berkaitan dengan penyimpanan data.
- 6) Jaringan komputer dan komunikasi data, yaitu sistem penghubung yang memungkinkan sumber (*resources*) dipakai secara bersama atau diakses oleh sejumlah pemakai.

Berdasarkan penjelasan diatas, penulis menyimpulkan bahwa sistem informasi memiliki komponen-komponen yang saling berinteraksi serta berhubungan dalam membentuk satu kesatuan untuk mencapai sasarannya.

c. Tujuan Sistem Informasi

Menurut (Sutabri, 2012), tujuan sistem informasi yaitu untuk menghasilkan produk informasi yang tepat bagi para pemakai akhir. Produk informasi meliputi pesan, laporan, formulir, dan gambar grafis, yang dapat disediakan melalui tampilan video, respon audio, produk kertas, dan multimedia.

C. E-Office

1. Definisi Electronic Office (E-Office)

Electronic Office (e-office) adalah suatu sistem yang berhubungan dengan administrasi secara maya memusatkan komponen-komponen sebuah organisasi dimana data, informasi dan komunikasi dibuat melalui media telekomunikasi (Ardhianto, 2015). Jadi dapat diartikan bahwa *e-office* adalah pekerjaan yang berhubungan dengan administrasi perkantoran seperti pengelolaan surat masuk, surat keluar dan disposisi surat yang dikerjakan menggunakan bantuan alat komunikasi dan sistem informasi.

E-Office adalah aplikasi yang digunakan untuk mempermudah pengelolaan surat masuk, surat keluar, disposisi dan pencarian arsip surat. Dengan fitur yang lengkap, akan sangat membantu mempermudah pengelolaan surat di instansi. Jika dulu pimpinan harus ada di kantor untuk melakukan disposisi, kini dengan sistem *e-office* pimpinan dapat langsung mendisposisikan surat secara *online* kepada staff yang ditugaskan untuk mengerjakan tugas tersebut. *E-office* juga dapat memfasilitasi pelacakan disposisi surat sehingga dapat mengetahui posisi surat dan perlu notifikasi pelaksanaan tugas tersebut.

E-Office merupakan aplikasi perkantoran yang mengganti proses administrasi manual ke proses berbasis elektronik dengan memanfaatkan fasilitas jaringan lokal (LAN), maupun jaringan internet (*online*). Dengan perkembangan telekomunikasi saat ini, jaringan broadband berkecepatan

tinggi, dan internet, maka perkantoran telah menjadi elektronis secara elektronik. Pekerjaan kantor yang selama beberapa dekade dilakukan secara manual telah berubah menjadi elektronik bahkan beberapa sudah menerapkan kantor maya. Kantor elektronis menggunakan teknologi komunikasi untuk menyediakan layanan administrasi perkantoran secara elektronis kepada siapa saja, dimana saja, dan tiap saat.

Adapun manfaat dari penggunaan *E-office* ini diantaranya, yaitu :

- a) Informasi akurat dan *realtime*.
- b) Proses bisnis *online* sehingga pengambilan keputusan cepat dan akurat.
- c) Pekerjaan tidak dibatasi ruang dan waktu sehingga produktivitas meningkat.
- d) Pelayanan lebih mudah dan lebih cepat untuk meningkatkan kepuasan pihak yang terlibat.
- e) Meningkatkan daya saing untuk pertumbuhan instansi.

2. Surat Masuk

Surat masuk adalah surat yang diterima dari perusahaan atau instansi lain kepada pihak yang bersangkutan. Surat masuk merupakan saran komunikasi tertulis yang diterima dari instansi atau perorangan.

Menurut (Sapto Aji, Migunani, 2014), surat masuk adalah semua jenis surat yang diterima dari organisasi/ instansi maupun perorangan, baik yang diterima melalui pos (kantor pos), maupun yang diterima dari kurir (pengantar surat) dengan mempergunakan buku pengiriman.

3. Disposisi

Disposisi menjadi kegiatan opsional setelah pendistribusian surat, dimana langkah-langkah pendistribusian surat adalah menyampaikan surat-surat ke alamat yang dituju, mengklasifikasikan surat dengan menyusun berdasarkan tingkat kepentingannya dan menentukan pejabat mana yang harus menangani surat dengan menyertakan lembar disposisi.

Disposisi merupakan petunjuk singkat tentang tindak lanjut (penyelesaian) terhadap suatu urusan atau surat. Disposisi adalah petunjuk tertulis mengenai tindak lanjut pengelolaan Surat bersama lembar disposisi diantarkan oleh kurir ke dinas atau biro yang dituju. Kemudian apabila diperlukan pejabat yang berwenang dapat melakukan disposisi lanjutan kepada bawahannya hingga surat sampai ke tangan pelaksana untuk ditindaklanjuti. Setelah surat ditindaklanjuti, maka pelaksana akan memberikan laporan kepada pimpinan yang telah memberikan disposisi (Sapto Aji, Migunani, 2014).

D. Model View Controller (MVC)

Model View Control (MVC) pertama kali diperkenalkan peneliti Xerox PARC yang bekerja pada bahasa pemrograman Smalltalk di akhir 1970-an dan awal 1980-an. Smalltalk adalah bahasa pemrograman yang berorientasi objek, bertipe dinamis, dan reflektif. Smalltalk pertama kali digunakan dalam pembelajaran edukasi dan hal ini berbeda dari data mainframe dan struktur control dalam program Smalltalk yang terlibat pada *Windowed User Interfaces*, konsep pemrograman berorientasi objek, pengantar pesan antara komponen-komponen objek, dan kemampuan untuk memonitor dan memodifikasi struktur dan perilakunya sendiri (Setiadi & Alfiah, 2016).

Model View Controller (MVC) merupakan model pembuatan program dengan arsitektur yang memisahkan proses, tampilan serta penghubung proses dan tampilan (Badiyanto, 2013). MVC merupakan sebuah *pattern* atau teknik pemrograman yang memisahkan antar pengembang aplikasi berdasarkan komponen utama pada sebuah aplikasi, seperti manipulasi data, *user interface* dan bagian yang menjadi kontrol aplikasi (Setiadi & Alfiah, 2016). Pengertian MVC adalah sebuah bentuk pemrograman yang memisahkan berdasarkan logika penanganan tampilan, logika pengontrolan dan logika model supaya pada pengembangan perangkat lunak yang besar mudah untuk dilakukan maintenance (Pasaribu et al., 2017).

Tujuan dari penerapan MVC adalah untuk menyederhanakan *script* program dan memudahkan *programmer* dalam memelihara sistem, karena data (*model*) terpisah dari tampilan (*view*) juga pengolahannya (*controller*), sehingga *programmer* tidak kesulitan saat memperbaiki bug pada sistem, atau bahkan melakukan pengembangan sistem lanjutan (Sudradjat et al., 2022). Menurut (M. Shalahuddin, 2013), konsep *Model View Controller* (MVC) bertujuan agar sebuah aplikasi dapat mudah dipelihara oleh orang-orang di dalam tim pengembang yang berbeda spesifikasi pekerjaan, misalnya *database administrator* (DBA) untuk mengurus masalah basis data, blok *controller* untuk *programmer*, dan blok *view* untuk desainer antarmuka (*interface designer*).

Menurut (Pasaribu et al., 2017) metode MVC memiliki tiga bagian, yaitu :

1. *Model*

Model adalah bagian kode program yang menangani *query* atau *database*. Jadi isi dari *model* merupakan bagian (fungsi-fungsi) yang berhubungan langsung dengan *database* untuk memanipulasi data seperti memasukkan data, pembaruan data, hapus data, dan lain-lain, namun tidak dapat berhubungan langsung dengan bagian *view*.

2. *View*

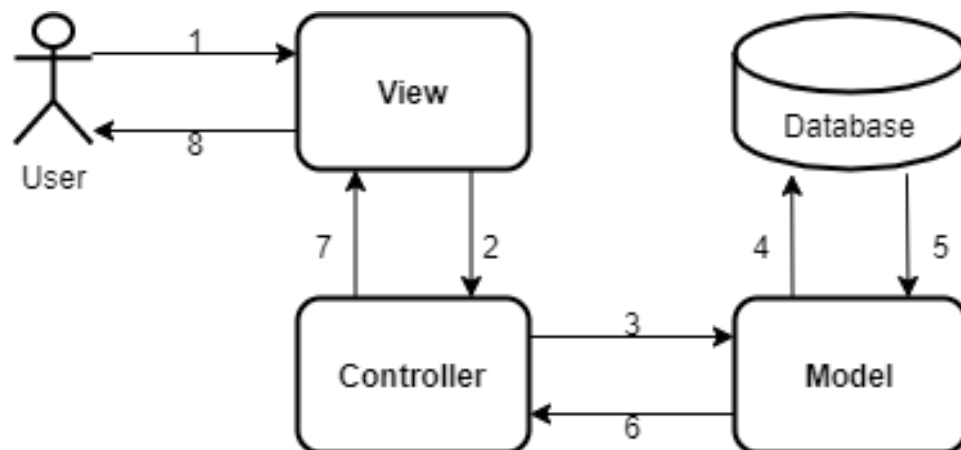
View adalah bagian kode program yang mengatur tampilan *website*. Pada aplikasi web bagian *view* biasanya berupa file *template* HTML, yang diatur oleh *controller*. Bagian ini tidak memiliki akses langsung terhadap bagian *model* namun berhubungan langsung dengan *controller*. *View* berfungsi untuk menerima dan merepresentasikan data kepada pengguna. Jadi bisa di katakan bahwa *view* merupakan halaman web.

3. *Controller*

Controller merupakan bagian yang menjembatani *model* dan *view*. *Controller* berisi perintah-perintah yang berfungsi untuk memproses suatu data dan mengirimkannya ke halaman web. *Controller* berfungsi untuk

menerima *request* dan data dari user kemudian menentukan apa yang akan diproses oleh aplikasi.

Adapun *Model View Controller* (MVC) memiliki alur kerja, yaitu sebagai berikut :



Gambar 2. 4 Alur Kerja MVC

- 1) User akan berinteraksi dengan tampilan pada bagian *interface* yaitu *view*.
- 2) Ketika user merequest data melalui *user interface*, *view* akan meneruskan *request* data tersebut kepada *controller*.
- 3) *Controller* menerima *request* data dari *view* lalu meneruskan *request* tersebut kepada *model*, *controller* berfungsi sebagai penghubung antara *view* dan *model*.
- 4) *Model* mengolah dan mencari data yang dibutuhkan pada *database*.
- 5) Setelah data ditemukan selanjutnya *model* akan mengirimkan data tersebut kepada *controller* untuk ditampilkan pada *view*.
- 6) *Controller* akan mengambil data hasil pengolahan *model* kemudian menampilkan kepada user melalui *view* halaman web.

Menurut (Rahmadiansyah, Dudi & Irwan, 2012) metode MVC memiliki kelebihan-kelebihan diantaranya, yaitu :

- 1) Bagian *model* memberikan penempatan detil data yang terpisah dan tidak disebar di dalam keseluruhan aplikasi sehingga meningkatkan kecepatan dan fleksibilitas dalam proses pemeliharaan aplikasi.
- 2) Pemisahan *model* juga membuat objek *model* dapat digunakan oleh aplikasi lain dengan kebutuhan yang sama (*reuse*).

- 3) Pemisahan *view* memudahkan perakitan/integrasi dengan komponen aplikasi lainnya tanpa harus memperhatikan detail proses. Desainer hanya berkonsentrasi pada bentuk dan tampilan antarmuka pemakai (*user interface*).
- 4) Penggunaan *Controller* memungkinkan untuk perubahan proses tanpa harus mengganggu antarmuka pemakai. Detail proses disembunyikan oleh *Controller* sehingga tidak mengganggu presentasi ke user maupun pengelolaan data/informasi (manajemen *database*).

Menurut (Rahmadiansyah, Dudi & Irwan, 2012) disamping kelebihan-kelebihannya, metode MVC juga memiliki kekurangan, yaitu :

- 1) Peningkatan kompleksitas aplikasi karena arsitektur aplikasi yang terbagi menjadi tiga bagian.
- 2) Menimbulkan *loosely coupled* (komponen aplikasi yang membuat/mengacu ke komponen lainnya dengan sedikit/tanpa informasi detail dari komponen yang diacu tersebut).

E. Alat Bantu Pengembangan Sistem

Untuk dapat mengikuti langkah-langkah pengembangan sistem sesuai dengan metodologi pengembangan sistem yang terstruktur, maka dibutuhkan alat dan teknik untuk melakukannya. Alat yang digunakan dalam perancangan sistem umumnya berupa gambar dan diagram. Alat yang digunakan dalam perancangan sistem yang digunakan dalam penelitian ini adalah :

1. *Unified Modeling Language (UML)*





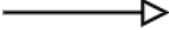
Menurut (M. Shalahuddin, 2013), UML adalah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks pendukung. UML merupakan sebuah standarisasi pemodelan perangkat lunak yang dibangun dengan teknik pemrograman berorientasi.

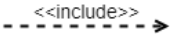
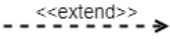

Kemudian dijelaskan bahwa UML sendiri memiliki komponen-komponen yang berupa diagram. Berikut masing-masing diagram tersebut:

a. Use Case Diagram

Use case diagram merupakan pemodelan untuk menggambarkan kelakuan aktor terhadap sistem yang akan dibuat. Simbol-simbol yang ada pada *Use case diagram* dapat dilihat pada tabel 2.2.

Tabel 2. 2 *Use Case Diagram*

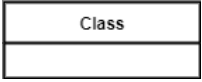
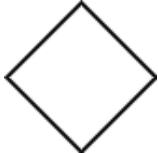
Simbol	Nama	Keterangan
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>Independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>Independent</i>).
	<i>Generalization</i>	Hubungan dimana objek anak (<i>Descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>Ancestor</i>).



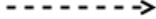

	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Assosiation</i>	Menghungkan antara objek satu dengan objek lainnya.

b. *Class Diagram*

Class diagram merupakan gambaran struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Class diagram* dibuat agar pembuat program membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Simbol- simbol yang ada pada diagram kelas dapat dilihat pada tabel 2.3.

Tabel 2. 3 *Use Case Diagram*


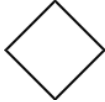



Simbol	Nama	Keterangan
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.

	<i>Asosiasi</i>	Hubungan statis antar <i>class</i> yang menggambarkan <i>class</i> yang memiliki atribut berupa <i>class</i> lain atau <i>class</i> yang harus mengetahui eksistensi <i>class</i> lain.
	<i>Generalization</i>	Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor).
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (independent).
	<i>Aggreration</i>	Hubungan yang lebih khusus dari Assosiasi dimana sebuah object memiliki lifecycle nya sendiri tapi dengan kepemilikan dan class child tidak dapat memiliki class parentnya. Relasinya biasa di sebut relasi “has-a”.

c. *Activity Diagram*

Activity diagram merupakan gambaran aliran kerja atau aktifitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Simbol-simbol yang ada pada activity diagram dapat dilihat pada tabel 2.4.


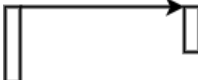

Tabel 2. 4 *Use Case Diagram*

Simbol	Nama	Keterangan
	<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antar muka saling berinteraksi satu sama lain.
	<i>Decision</i>	Pilihan untuk pengambilan keputusan.
	<i>Initial Node</i>	Titik Awal.
	<i>Activity Final Node</i>	Titik akhir.
	<i>Fork Node</i>	Menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.

d. *Sequence Diagram dan Collaboration Diagram*

Sequence diagram menggambarkan interaksi antara objek di dalam dan disekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu, menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu. *Collaboration diagram* juga menggambarkan interaksi antara objek seperti *sequence diagram*, akan tetapi lebih menekankan pada masing-masing objek dan bukan pada waktu penyampaian *message*. Setiap *message* memiliki *sequence number*, dimana *message* dari level tertinggi memiliki nomor 1 (Dharwiyanti, 2003). Simbol-simbol yang ada pada *sequence diagram* dapat dilihat pada tabel 2.5.



Tabel 2. 5 *Sequence Diagram dan Collaboration Diagram*

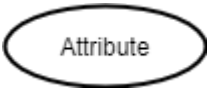

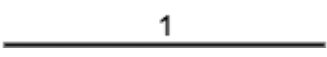
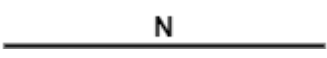
Simbol	Nama	Keterangan
	<i>Object dan Lifeline</i>	Orang, tempat, benda, kejadian atau konsep yang ada dalam dunia nyata yang penting bagi suatu aplikasi yang saling berinteraksi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi.
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .

2. *Entity Relationship Diagram (ERD)*

Menurut (Jogiyanto, 2005), *Entity Relationship Diagram (ERD)* adalah bagian yang menunjukkan hubungan antara *entity* yang ada dalam sistem. Simbol-simbol *Entity Relationship Diagram* yang digunakan dapat dilihat pada tabel 2.6.

Tabel 2. 6 Entity Relationship Diagram

Simbol	Keterangan
	Entitas
	<i>Relasi</i>

	<i>Atribut</i>
	<i>Atribut kunci</i>
	Hubungan satu <i>entitas</i>
	Hubungan banyak <i>entitas</i>

F. Perangkat Lunak Pembangun Sistem

1. PHP

a. Sejarah PHP

PHP pertama kali dikembangkan oleh Rasmus Lerdorf pada tahun 1994. Pada awalnya PHP merupakan kependekan dari *Personal Home Page* (Situs personal). Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web.

Pada tahun 1995, sebuah perusahaan bernama Zend menulis ulang *interpreter* PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1995, perusahaan tersebut merilis *interpreter* baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP diubah menjadi akronim berulang PHP: *Hypertext Preprocessor*.

Pada pertengahan tahun 1999, Zend merilis *interpreter* PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari *interpreter* PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek. Peladen web bawaan ditambahkan pada versi 5.4 untuk mempermudah pengembang menjalankan kode PHP tanpa menginstal peladen perangkat lunak.

Versi terbaru dan stabil dari bahasa pemrograman PHP saat ini adalah versi 8.0.

b. Definisi PHP

(Arief, 2011) mengatakan bahwa *Personal Home Page* (PHP : *Hipertext Preprocessor*) adalah bahasa *server side scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Karena PHP merupakan *server side scripting* maka sintaks dan perintah-perintah PHP akan dieksekusi di *server* kemudian hasilnya dikirimkan ke *browser* dalam format HTML.

c. Kelebihan PHP

Bahasa pemrograman PHP merupakan bahasa pemrograman yang paling banyak digunakan, karena berbagai alasannya, salah satu adalah mempunyai beberapa kelebihan dibandingkan dengan Bahasa pemrograman yang lain atau sejenisnya. Berikut ini kelebihan bahasa pemrograman PHP :

- 1) PHP adalah bahasa *multiplatform* yang artinya dapat berjalan di berbagai mesin atau sistem operasi (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem lainnya.
- 2) PHP bersifat *Open Source* yang berarti dapat digunakan oleh siapa saja secara gratis.
- 3) *Web Server* yang mendukung PHP dapat ditemukan di mana-mana dari mulai apache, IIS, Lighttpd, nginx, hingga Xitami dengan konfigurasi yang relatif mudah dan tidak berbelit-belit, bahkan

banyak yang membuat didalam bentuk paket atau package (PHP, MySQL dan Web Server).

- 4) Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis, komunitas dan *developer* yang siap membantu dalam pengembangan.
- 5) Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
- 6) Banyak bertebaran Aplikasi dan Program PHP yang Gratis dan Siap pakai seperti WordPress, PrestaShop, dan lain-lain.
- 7) Dapat mendukung banyak *database*, seperti MySQL, Oracle, MS-SQL dst.

d. *Sintax/Script PHP*

Script PHP menurut (Dewi et al., 2021) termasuk dalam *HTML-embedded*, artinya kode PHP dapat disisipkan pada sebuah halaman HTML. Ada empat macam pasangan tag PHP yang dapat digunakan untuk menandai blok *script* PHP, yaitu :

- 1) `<?php...?>`
- 2) `<script language= "PHP">...</script>`
- 3) `<? ... ?>`
- 4) `<% ... %>`

2. *PhpMyAdmin*

PhpMyAdmin pertama kali didirikan oleh Tobias Ratschiller. Saat itu, Ratschiller adalah seorang konsultan IT. Ratschiller juga kemudian mendirikan perusahaan perangkat lunak Maguma.

Ratschiller mulai mengerjakan front-end berbasis PHP ke *MySQL* pada tahun 1998, terinspirasi oleh *MySQL-Webadmin* Peter Kuppelwieser. Dia sempat berhenti mengerjakan proyek tersebut dan juga *phpAdsNew* pada tahun 2000 karena kurangnya waktu.

Pada saat itu, *phpMyAdmin* sudah menjadi salah satu aplikasi PHP dan alat administrasi *MySQL* yang paling populer, dengan komunitas pengguna dan kontributor yang besar.

Untuk mengkoordinasikan semakin banyak *patches*, tiga orang pengembang, Olivier Müller, Marc Delisle dan Loïc Chapeaux, mendaftarkan proyek *phpMyAdmin* di SourceForge.net dan mengambil alih pembangunan pada tahun 2001. Sejak tahun 2015 pembangunan sepenuhnya didasarkan pada GitHub.

Adapun fitur-fitur dari *phpMyAdmin* yaitu :

- *phpMyAdmin* memiliki *interface* yang *user-friendly* dan intuitive yang mudah dipelajari. Dengan begitu, orang-orang dapat mempelajari dan membiasakan diri dengan fitur-fiturnya dengan mudah dan cepat.
- *phpMyAdmin* memperbolehkan user memanfaatkan kebanyakan fungsi *MySQL*, termasuk mengelola *database*, menjalankan *queries MySQL*, mengeksekusi *statement MySQL*, mengelola user dan *permission* dan masih banyak lagi.
- Anda juga bisa mengimport dan mengexport data dari dan ke berbagai format.
- Dengan *pre-defined fuctions*, Anda bisa mengubah format data-data yang sudah tersimpan.
- *phpMyAdmin* bisa membuat *graphics database* dalam bentuk PDF, bersamaan dengan beberapa *queries* yang kompleks dibantu dengan *query-by-example*.
- Melakukan pencarian pada *database*.

3. *Mysql*

MySQL merupakan *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis di bawah lisensi GPL (General Public Liscence). Dimana setiap orang bebas untuk menggunakan *MySQL*, namun tidak boleh dijadikan produk turunan yang bersifat *closed source* atau komersial.

MySQL sebenarnya merupakan turunan salah satu konsep utama dalam *database* sejak lama, yaitu SQL (*Structure Query Language*). SQL merupakan sebuah konsep pengoperasian *database*, terutama untuk pemilihan/seleksi dan pemasukkan data yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Kendala suatu sistem *database* (DBMS) dapat diketahui dan cara kerja *optimizernya* dalam melakukan proses perintah-perintah SQL yang dibuat oleh user maupun program-program aplikasinya.

Sebagai *database server*, *MySQL* dapat dikatakan lebih unggul dibandingkan dengan *database server* yang lainnya dalam *query* data. Adapun *MySQL* memiliki banyak sekali keistimewaan yang dimilikinya, yaitu :

1) *Portability*

MySQL dapat berjalan stabil pada berbagai sistem operasi diantaranya adalah seperti Windows, Linux, FreeBSD, Mac OS X server, Solaris, Amiga, HP-UX dan masih banyak lagi.

2) *Open Source*

MySQL didistribusikan secara *open source* (gratis), dibawah lisensi GPL.27.

3) *Multiuser*

MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik. Hal ini memungkinkan sebuah *database server MySQL* dapat diakses *client* secara bersamaan.

4) *Perfomance Tuning*

MySQL memiliki kecepatan yang menabjukkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5) *Column Types*

MySQL memiliki tipe kolom yang sangat kompleks, seperti *signed/unsigned integer, float, double, char, varchar, text, blob, date, time, datetime, year*, serta *enum*.

6) *Command and Function*

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *SELECT* dan *WHERE* dalam query.

7) *Security*

MySQL memiliki beberapa lapisan sekuritas seperti level *subnetmask*, nama *host*, dan user dengan sistem perizinan yang mendetail serta *password* terenkripsi.

8) *Stability dan Limits*

MySQL mampu menangani *database* dalam skala besar, dengan jumlah *records* lebih dari 50 juta dan 60 ribu *table* serta 5 miliar baris. Selain itu, batas *indeks* yang dapat ditampung mencapai 32 *indeks* pada tiap tabelnya.

9) *Connectivity*

MySQL dapat melakukan koneksi dengan *client* menggunakan *protocol TCP/IP, Unix socket (Unix)*, atau *Named Pipes (NT)*.²⁸

10) *Localisation*

MySQL dapat mendeteksi pesan kesalahan (*error code*) pada *client* dengan menggunakan lebih dari dua puluh bahasa. Meski demikian, bahasa Indonesia belum termaksud di dalamnya.

11) *Interface*

MySQL memiliki *interface* (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).

12) *Client dan Tools*

MySQL dilengkapi dengan berbagai *tool* yang dapat digunakan untuk administrasi *database*, dan pada setiap *tool* yang ada disertai petunjuk *online*.

13) *Struktur Tabel*

MySQL memiliki stuktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE*, dibandingkan dengan *database* lainnya semacam PostgreSQL ataupun Oracle.

4. XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), *Apache*, *MySQL*, PHP dan *Perl*. Program ini tersedia dalam GNU *General Public License* dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman web dinamis.

5. Framework Codeigniter

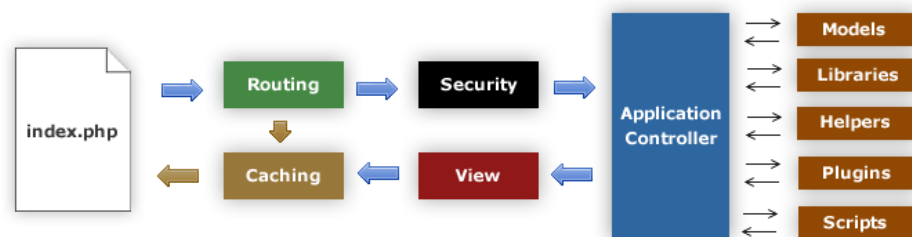
Framework adalah abstraksi di dalam sebuah perangkat lunak yang menyediakan fungsi yang *generic* sehingga dapat dirubah oleh kode yang dibuat user sehingga dapat menyediakan perangkat lunak untuk aplikasi tertentu.

Metode MVC adalah sebuah arsitektur yang dapat diimplementasikan secara bebas dengan atau tanpa bahasa pemrograman berorientasi objek. Dengan demikian metode MVC dapat diimplementasikan dalam sebuah *framework*.

CodeIgniter merupakan sebuah *framework* pemrograman web dengan menggunakan bahasa *php*. *Framework* ini ditulis dengan menggunakan bahasa *php* versi 4 dan versi 5 oleh Rick Ellislab yang menjadi CEO Ellislab, Inc. dan dipublikasikan dengan lisensi di bawah *Apache/BSD Open Source*. Jadi *CodeIgniter* adalah *framework php* dan *Open Source* (Rahmadiansyah, Dudi & Irwan, 2012).

Prinsip kerja utama *framework CodeIgniter* terletak pada file *index.php* yang diletakkan pada direktori *root* aplikasi. File ini akan memicu dan mengarahkan permintaan layanan halaman web ke dalam tubuh *framework CodeIgniter*.

Menurut (Rahmadiansyah, Dudi & Irwan, 2012) mekanisme kerja *CodeIgniter*, yaitu :



Gambar 2. 5 Alur kerja *CodeIgniter*

- 1) File *index.php* bertindak sebagai pengendali utama yang berfungsi memuat kode *script* utama yang berfungsi menjalankan *CodeIgniter*.
- 2) Selanjutnya, modul *routing* berfungsi menerima permintaan layanan HTTP untuk menentukan arah eksekusi *script* yang akan dilaksanakan.
- 3) Jika konfigurasi *cache* tersedia, maka sistem langsung mengeksekusi untuk ditampilkan di halaman web.
- 4) Permintaan-permintaan layanan HTTP dan data-data dari form yang dikirimkan ke *server*, akan di-*filter* dan diamankan oleh modul *security*.
- 5) Selanjutnya data dikirimkan ke modul *Controller* yang akan mengendalikan akses ke modul *Model*, *Library*, *Helper*, *Plugin*, dan modul-modul sumber daya lainnya.
- 6) Kemudian *Controller* akan mengirimkan data ke modul *View* untuk ditampilkan ke halaman web. Jika konfigurasi *caching* diaktifkan, maka sebelum tampilan ini dikirimkan ke web untuk ditampilkan ke *browser*, maka tampilan ini akan di-*cache* sehingga permintaan yang sama dapat dilakukan dengan lebih cepat.

Adapun manfaat dari penggunaan *framework CodeIgniter* menurut (Rahmadiansyah, Dudi & Irwan, 2012) adalah :

- 1) Gratis, sesuai dengan semangat *Open Source* untuk dapat digunakan dan dikembangkan secara bersama-sama. Dapat di-*download* pada alamat <http://CodeIgniter.com/downloads/> secara gratis, bebas digunakan sesuai persyaratan persetujuan lisensi (*licence agreemen*) yang bisa dilihat pada *website* tersebut di atas.
- 2) Ditulis dengan menggunakan bahasa *php* 4 (untuk versi 1.x.x) dan versi 5 (untuk versi 2.x.x) sehingga mendukung pemrograman dengan bahasa *php*.
- 3) Menggunakan metode MVC sebagai prinsip kerjanya sehingga dapat digunakan untuk mengembangkan aplikasi secara efisien dan dinamis serta lebih memudahkan dalam melakukan pemeliharaan aplikasi.
- 4) Menggunakan URL (*Uniform Resource Locator*) yang sederhana, bersih, dan SEF (*Search Engine Friendly*).
- 5) Memiliki paket *library* yang lengkap, mendukung fungsi-fungsi *database, html, web, e-mail, session, pagination* dan lain-lain.
- 6) Dokumentasi yang lengkap dan jelas, disertakan dalam *website* resminya dan dapat di-*download* bersama-sama dengan *framework*-nya.
- 7) Komunitas, *framework* ini didukung oleh banyak pengguna dan pengembang, walaupun awalnya dikembangkan oleh Ellislab, Inc.
- 8) Bersifat *portabel* dan dapat dijalankan pada berbagai *platform* yang mendukung bahasa pemrograman *php*.

Disamping kelebihan-kelebihan diatas, *framework CodeIgniter* juga memiliki kelemahan-kelemahan, yaitu (Rahmadiansyah, Dudi & Irwan, 2012) :

- 1) Longgar dalam penerapan aturan MVC, sehingga pemrograman masih diberikan kesempatan untuk melanggar kaidah-kaidah MVC.
- 2) Tidak mendukung konsep ORM (*Object Relational Model*) yaitu metode pengaksesan *database* dengan menggunakan relasi antar objek sehingga pemrogram tidak perlu menuliskan atau mengetahui sintaks bahasa SQL.

- 3) Walaupun dikembangkan oleh komunitas, namun jumlah pengembangnya tidak sebesar *framework php* lainnya seperti CakePHP, karena masih di bawah koordinasi Ellislab, Inc, sehingga *update core engine*-nya lebih lama daripada *framework Open Source* lainnya.
- 4) Sebagai *framework Open Source*, *CodeIgniter* tidak menyediakan dukungan (*support*) secara khusus kecuali melalui forum pengguna.

6. *Visual Studio Code*

Visual Studio Code merupakan sebuah aplikasi editor kode *open source* yang dikembangkan oleh *Microsoft* untuk sistem operasi *Windows*, *Linux*, dan *MacOS*. *Visual Code* memudahkan dalam penulisan kode yang mendukung beberapa jenis pemrograman, seperti C++, C#, *Java*, *Python*, PHP, GO. *Visual Code* memiliki kemampuan untuk mengidentifikasi jenis bahasa pemrograman yang digunakan dan memberi variasi warna sesuai dengan fungsi dalam rangkaian code tersebut. *Visual Studio Code* juga telah terintegrasi ke *Github*. Selain itu fitur lainnya adalah kemampuan untuk menambah ekstensi dimana para pengembang dapat menambah ekstensi untuk menambah fitur yang tidak ada di *Visual Studio Code* (Ramdhan & Nufriana, 2019).

BAB III

ANALISA DAN HASIL

A. Analis Sistem

Analisis sistem adalah penguraian dari suatu sistem yang utuh ke berbagai bagian komponennya dengan tujuan agar peneliti dapat mengidentifikasi dan mengevaluasi berbagai masalah atau hambatan yang muncul pada sistem sehingga nantinya dapat dilakukan penanggulangan, perbaikan dan pengembangan.

Tahap analisa sistem dilakukan sebelum tahap perancangan sistem. Tujuan analisa terhadap yang lama adalah untuk mendapatkan suatu hasil yang lebih baik dengan melihat beberapa kelemahan dan kekurangan dari sistem yang lama. Permasalahan-permasalahan yang ditemukan pada saat analisa sistem dapat diatasi dengan mengadakan perubahan-perubahan dan merancang suatu sistem baru, dimana sistem baru tersebut merupakan perbaikan dari sistem lama yang tujuannya dapat menutupi kelemahan-kelemahan yang terjadi pada sistem lama.

1. Analisis Sistem Yang Sedang Berjalan

Analisa sistem yang sedang berjalan merupakan gambaran atau acuan untuk merancang sistem baru, karena dengan menganalisa sistem yang sedang berjalan akan diketahui kelemahan-kelemahan dari sistem yang lama dan keunggulan sistem baru. Sistem lama akan dijadikan perbandingan terhadap sistem baru yang akan diterapkan. Analisa sistem bertujuan mencari pemecahan masalah yang dihadapi sistem tersebut agar masalah yang lama tidak terjadi lagi dimasa yang akan datang.

Berikut gambaran sistem informasi pengolahan data surat menyurat di UIN Mahmud Yunus Batusangkar :

- a. Bagian resepsionis menerima surat masuk dan memberikan surat beserta lembar diposisi secara manual ke Kabagi umum.

- b. Kabag umum akan mengisi disposisi pada lembar disposisi untuk pimpinan selanjutnya (contohnya ke kabiro).
- c. Selanjutnya surat-surat akan dilanjutkan ke bagian disposisi selanjutnya sesuai dengan perintah disposisi.
- d. Setelah selesai proses disposisi maka akan dibuatkan jawaban surat dari surat-surat yang masuk tersebut.
- e. Terakhir lembar disposisi akan disimpan oleh resepsionis sebagai laporan disposisi.

2. Evaluasi Sistem Yang Sedang Berjalan

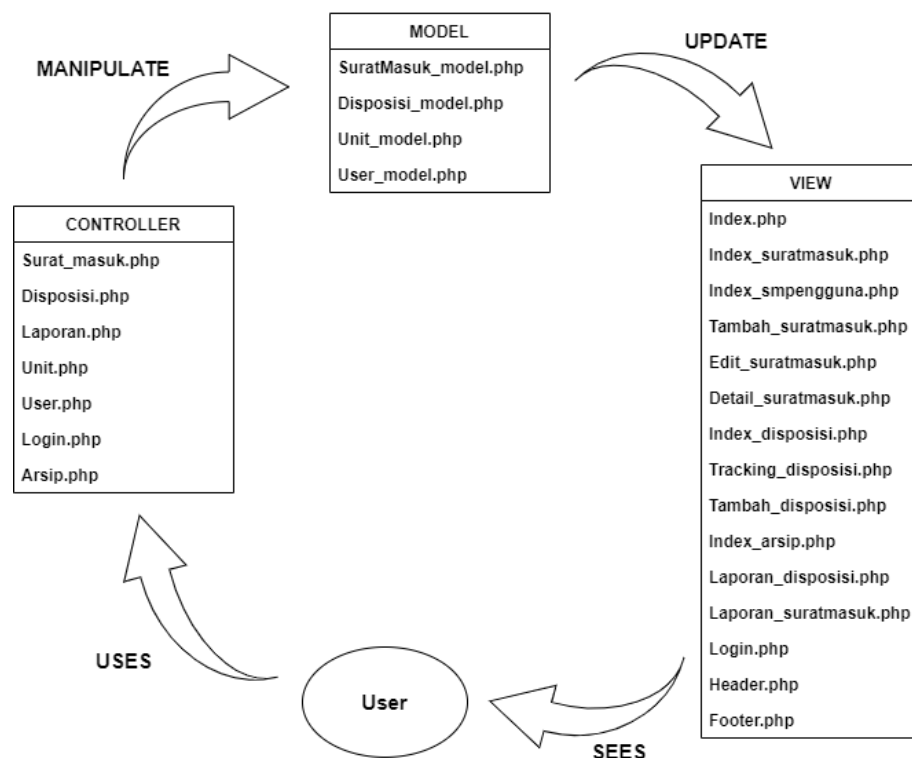
Evaluasi sistem dilakukan untuk mengetahui masalah yang terjadi pada sistem yang lama sebagai dasar untuk merancang sistem yang baru. Sistem yang ada sudah dapat dikatakan berjalan dengan baik, tetapi masih terdapat kekurangan dalam proses administrasi surat di lingkup UIN Mahmud Yunus Batusangkar, antara lain:

- a. Proses pengelolaan surat masuk dan disposisi surat masih dilakukan secara manual yaitu dengan menggunakan lembar disposisi yang diberikan langsung kepada pimpinan.
- b. Proses disposisi surat seperti ini seringkali mengalami penundaaan jika pimpinan sedang berada di luar kota atau tidak berada di kantor.
- c. Keberadaan posisi surat juga tidak dapat pantau secara realtime karena keberadaan posisi surat hanya dapat diketahui setelah surat dilakukan disposisi oleh pimpinan.
- d. Membutuhkan waktu yang lama untuk pencarian kembali data-data surat masuk yang telah berlalu ketika dibutuhkan.

3. Penerapan Konsep MVC pada *E-Office*

Untuk mengimplementasikan hasil rancangan kedalam sistem harus digunakan bahasa pemrograman dengan strategi dan logika yang tepat. Dalam kajian penelitian ini digunakan bahasa pemrograman PHP dengan *framework codeigniter* dengan strategi *Model View Controller*

(MVC). Dalam implementasi model MVC memisahkan antara data (*model*), tampilan/*interface* (*view*) dan program pengendali (*controller*). Setiap bagian dipisahkan dalam folder yang berbeda. Pada bagian controller berfungsi untuk memilah-milah request dari user, apakah akan menggunakan model atau hanya view saja dalam merespon request dari user, *Controller* juga berisi perintah-perintah yang berfungsi untuk memproses antarmuka. Sedangkan *Model* berhubungan dengan basis data dengan fungsi-fungsi yang berkaitan perintah memasukkan data, menghapus data, memperbaharui data dan lain-lain. Pada *View*, *view* adalah code yang berhubungan dengan website dan yang akan ditampilkan kepada user.



Gambar 3. 1 Konsep MVC pada E-Office

Pada Gambar 3.1 menunjukkan sebuah konsep MVC (*Model View Controller*) pada sistem E-Office, disana menjelaskan alur dari sebuah proses pemakaian konsep MVC dimana user akan menggunakan *Controller* untuk memanipulasi data yang kemudian data tersebut akan

diupdate ketika berada dibagian *Model* dan proses terakhir tugas dari *View* untuk menampilkan data tersebut kepada user.

Pada bagian *Controller*, File Surat_masuk.php yang dimana file ini berfungsi untuk memanipulasi data yang ada pada Surat Masuk. Selanjutnya File Disposisi.php yang berfungsi untuk memanipulasi data yang berhubungan dengan disposisi. File Laporan.php berfungsi untuk memanipulasi data berupa laporan surat masuk dan laporan disposisi surat. File Arsip.php berfungsi untuk memanipulasi data disposisi surat masuk yang diarsipkan. Dan pada File Login.php berfungsi untuk memanipulasi data yang berhubungan dengan keamanan dari sistem yaitu memberikan hak akses masuk sesuai dengan kegunaannya.

Pada bagian *Model*, terdapat File Suratmasuk_model.php yang dimana file ini berfungsi untuk memberikan pembaharuan data surat masuk yang telah dimanipulasi untuk dapat ditampilkan kepada user pada bagian *View*. File Disposisi.php berfungsi untuk memberikan pembaharuan data disposisi yang telah dimanipulasi untuk dapat ditampilkan kepada user pada bagian *View*.

Sedangkan pada bagian *View*, terdapat Folder Surat masuk yang berisikan beberapa file yang nantinya akan berfungsi untuk menampilkan data yang berhubungan dengan surat masuk kepada user. Folder Disposisi yang berisikan beberapa file yang nantinya berfungsi untuk menampilkan data yang berhubungan dengan disposisi kepada user. Folder Laporan yang terdiri dari file Laporan_suratmasuk.php dan Laporan_disposisi.php yang berfungsi untuk menampilkan *output* laporan kepada user. Dan Folder Login yang berisikan beberapa file yang berfungsi untuk menampilkan halaman login.

B. Perancangan Sistem

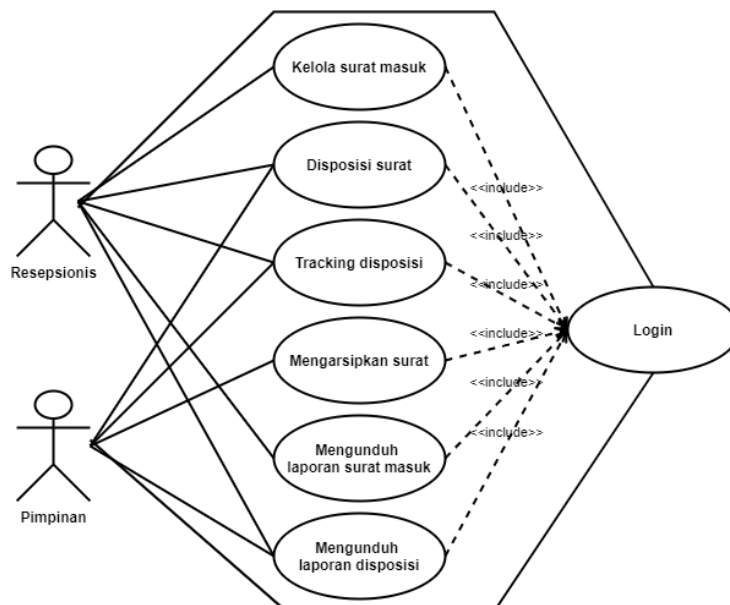
1. Perancangan Global

Perancangan global berfungsi untuk memudahkan dalam melakukan rancangan secara terinci, serta memberikan gambaran tentang hubungan antara sub-sub sistem. Perancangan global juga dapat memberikan kemudahan bagi pemakai dalam mempelajari dan menggunakan sistem. Berdasarkan analisis yang dilakukan sebelumnya, maka pemodelan yang digunakan yaitu *Unified Modelling Language* (UML) yang mencakup *Use Case Diagram*, *Class Diagram*, *Activity Diagram* dan *Sequence Diagram* serta *Entity Relationship Diagram* (ERD).

a. *Unified Modelling Language* (UML)

1) *Use Case Diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem, yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use Case Diagram* dari sistem yang dirancang dapat digambarkan seperti pada gambar berikut di bawah ini.



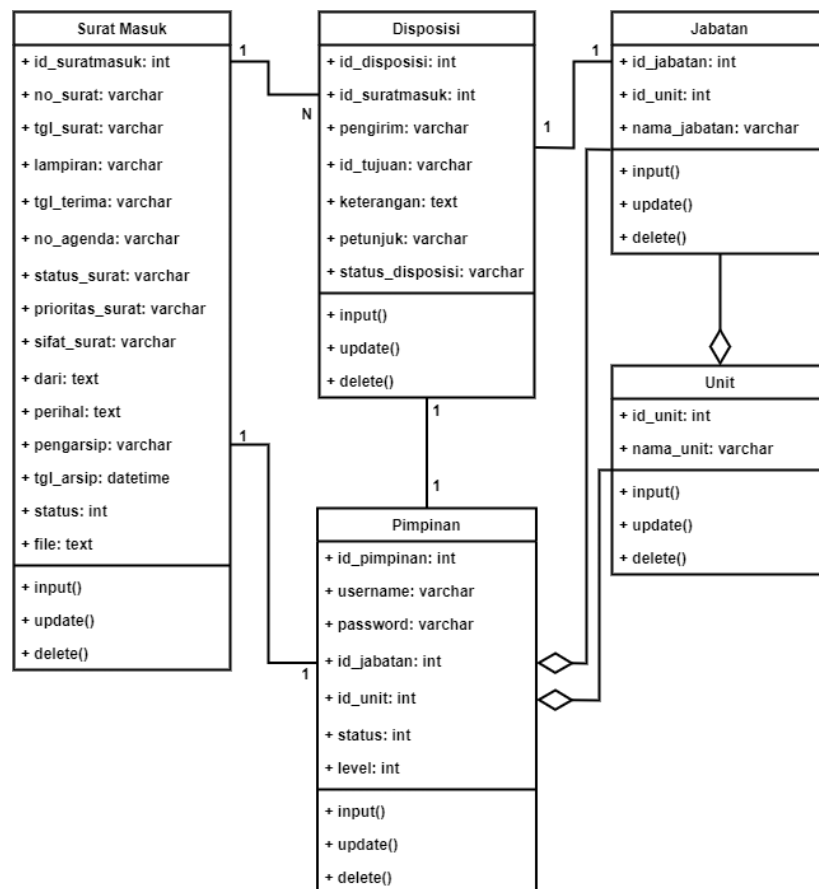
Gambar 3. 2 *Use Case Diagram*

Pada Gambar 3.2 Resepsionis harus melakukan login terlebih dahulu ke dalam aplikasi agar bisa mengelola surat masuk

(input, edit dan menghapus data surat), mengirimkan surat ke pimpinan melalui disposisi, tracking surat untuk mengetahui posisi surat serta mengunduh laporan surat masuk dan laporan disposisi surat. Pimpinan juga harus melakukan login terlebih dahulu ke dalam aplikasi agar dapat memberikan disposisi surat, mengarsipkan surat ketika surat sudah sampai pada penerima terakhir disposisi, tracking surat untuk mengetahui posisi surat serta mengunduh laporan disposisi surat.

2) *Class Diagram*

Class Diagram mendefenisikan informasi apa yang dimiliki suatu objek serta mendefinisikan perilaku yang dimilikinya. *Class diagram* mengabtraksikan elemen-elemen dari sistem sedang dibangun dan dirancang. *Class diagram* dari sistem yang dirancang dapat digambarkan seperti pada gambar berikut di bawah ini.



Gambar 3. 3 Class Diagram

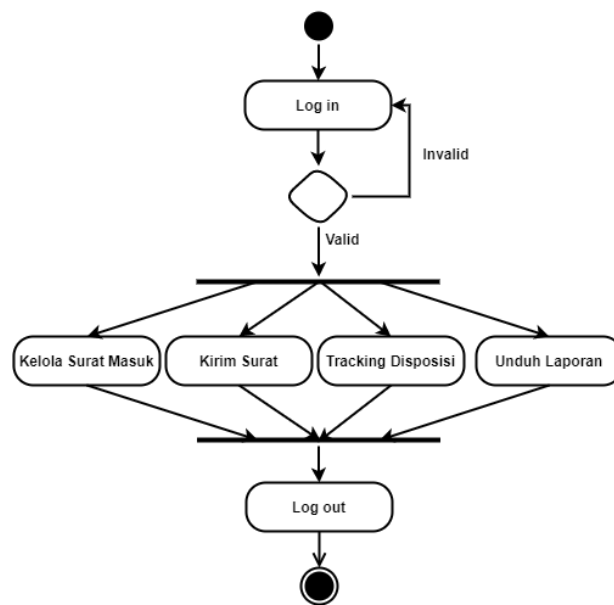
Pada Gambar 3.3 Class Surat Masuk akan memiliki banyak disposisi, setiap Class Pimpinan akan memiliki satu disposisi dan setiap Class Disposisi akan memiliki satu tujuan berdasarkan id jabatan. Setiap Class Surat Masuk nanti akan diarsipkan oleh satu pimpinan berdasarkan id_jabatan pimpinan. Class Pimpinan memiliki id_jabatan dan id_unit, dimana Class Jabatan juga memiliki satu id_unit.

3) Activity Diagram

Activity Diagram adalah gambaran visual dari alir kerja yang berisi aktivitas dan tindakan dari sistem. *Activity Diagram* menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, melainkan aktifitas yang dapat dilakukan oleh sistem.

a) Activity Diagram Resepsionis

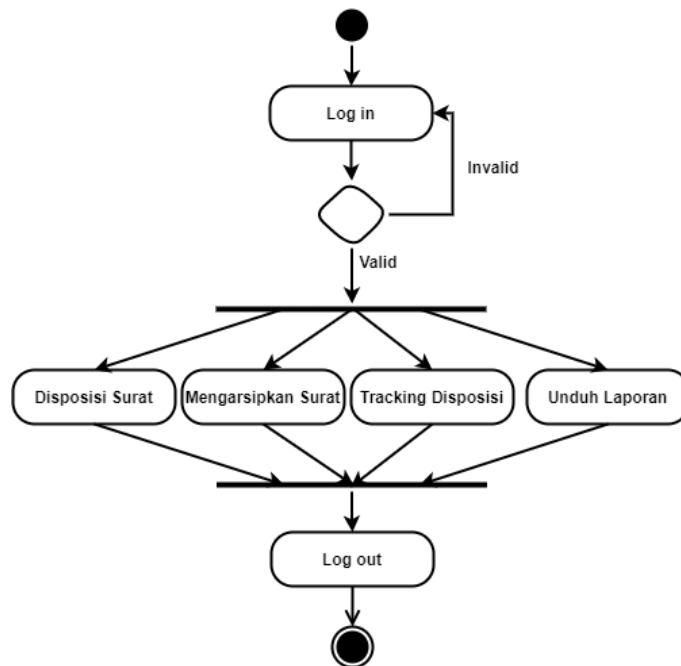
Activity diagram ini menggambarkan Resepsionis yang harus melakukan *login* terlebih dahulu agar bisa masuk ke dalam sistem dengan benar, Resepsionis dapat melakukan *input* data surat masuk, mengirim surat pada, *tracking* disposisi surat, dan mengunduh laporan surat masuk dan laporan disposisi surat yang telah diarsipkan.



Gambar 3. 4 *Activity Diagram* Resepsionis

b) *Activity Diagram* Pimpinan

Activity Diagram ini menggambarkan Pimpinan yang harus melakukan *login* terlebih dahulu agar bisa masuk ke dalam sistem dengan benar, Pimpinan dapat melakukan disposisi surat, mengarsipkan surat, *tracking* disposisi surat, dan mengunduh laporan disposisi surat yang telah diarsipkan.



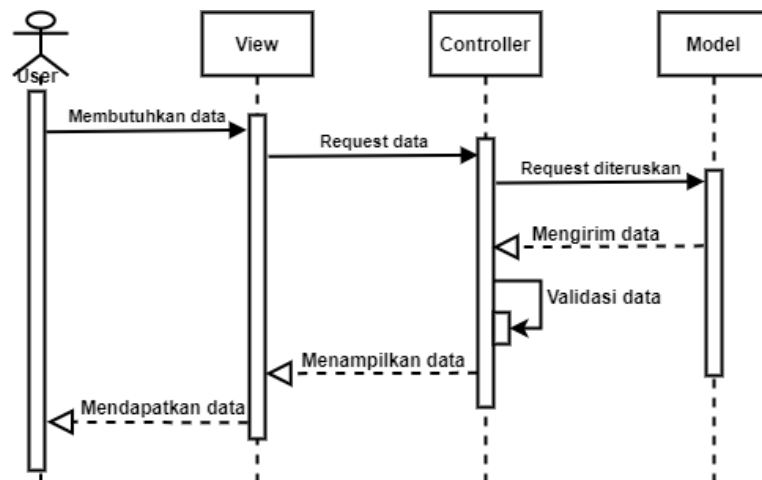
Gambar 3. 5 Activity Diagram Pimpinan

4) Sequence Diagram

Sequence Diagram menggambarkan interaksi antara objek di dalam dan di sekitar sistem, termasuk pengguna dan antar muka pengguna. *Sequence diagram* bisa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai *response* dari sebuah *event* untuk menghasilkan *output* tertentu.

a) Sequence Diagram MVC

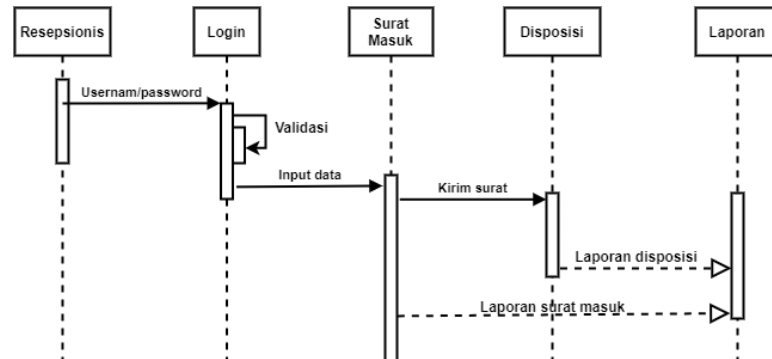
Sequence diagram MVC menggambarkan bagaimana metode *Model View Controller* menangani permintaan user hingga sampai pada *output* data yang ditampilkan pada Halaman web kepada user.



Gambar 3. 6 *Sequence Diagram MVC*

b) *Sequence Diagram* Resepsionis

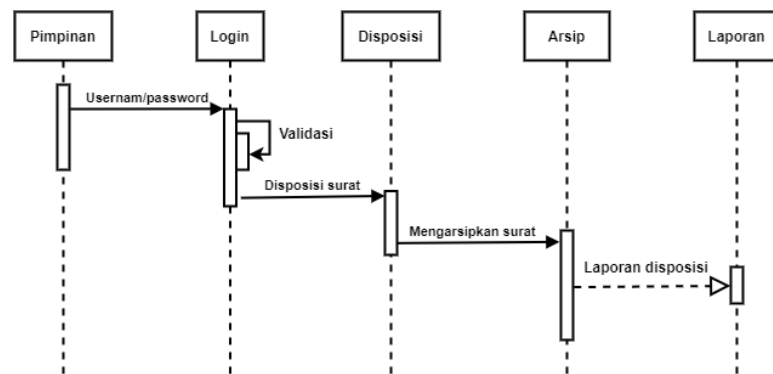
Sequence Diagram Resepsionis menunjukkan bagaimana Resepsionis dalam melakukan proses *login* ke dalam sistem hingga sampai *output* berupa laporan surat masuk dan laporan disposisi.



Gambar 3. 7 *Sequence Diagram Resepsionis*

c) *Sequence Diagram* Pimpinan

Sequence Diagram Pimpinan menunjukkan bagaimana pimpinan bisa melakukan proses *login* kedalam sistem hingga sampai *output* berupa laporan disposisi surat.

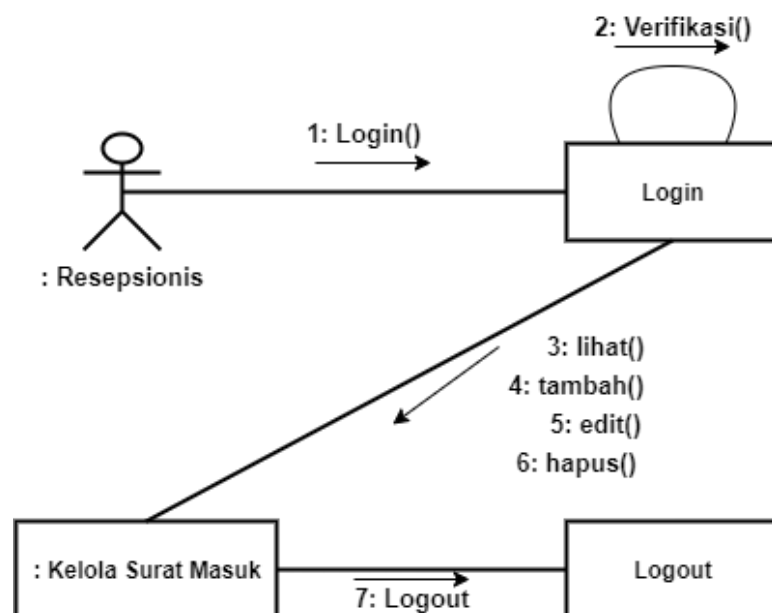


Gambar 3. 8 *Sequence Diagram Pimpinan*

5) *Collaboration Diagram*

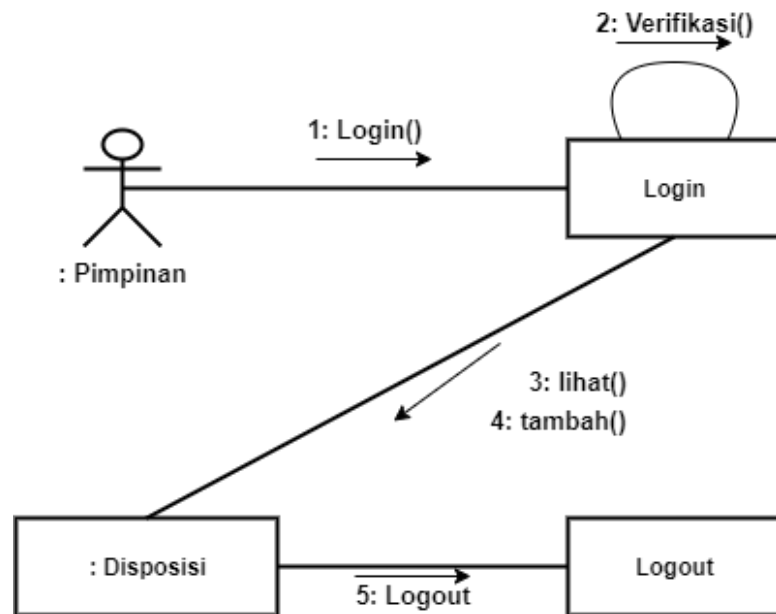
Collaboration diagram adalah cara alternatif untuk mengetahui tahap-tahap terjadinya suatu aktivitas. Perbedaan antara *collaboration* dan *sequence diagram* adalah *collaboration diagram* memperlihatkan bagaimana hubungan antara beberapa objek, sedangkan *sequence diagram* memperlihatkan bagaimana urutan kejadian.

a) *Collaboration diagram* Kelola Surat Masuk Resepsionis



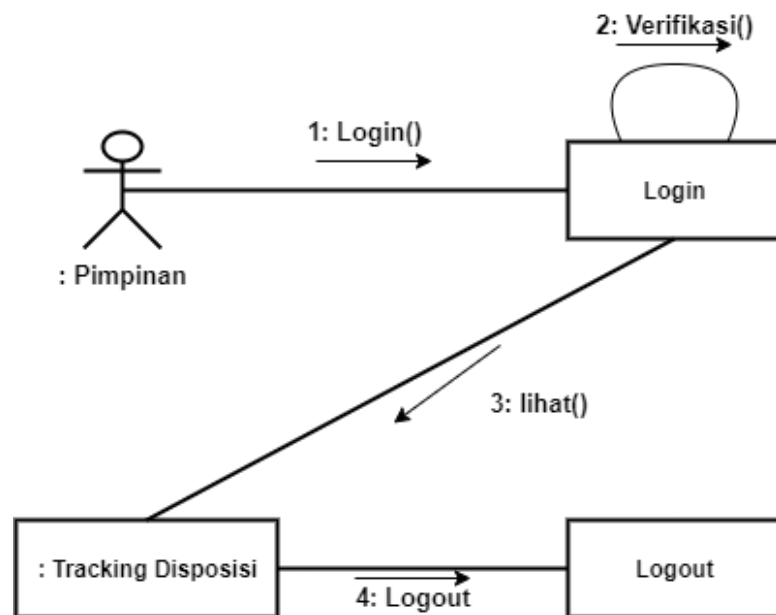
Gambar 3. 9 *Collaboration Diagram Kelola Surat Masuk Resepsionis*

b) *Collaboration diagram* Disposisi Pimpinan



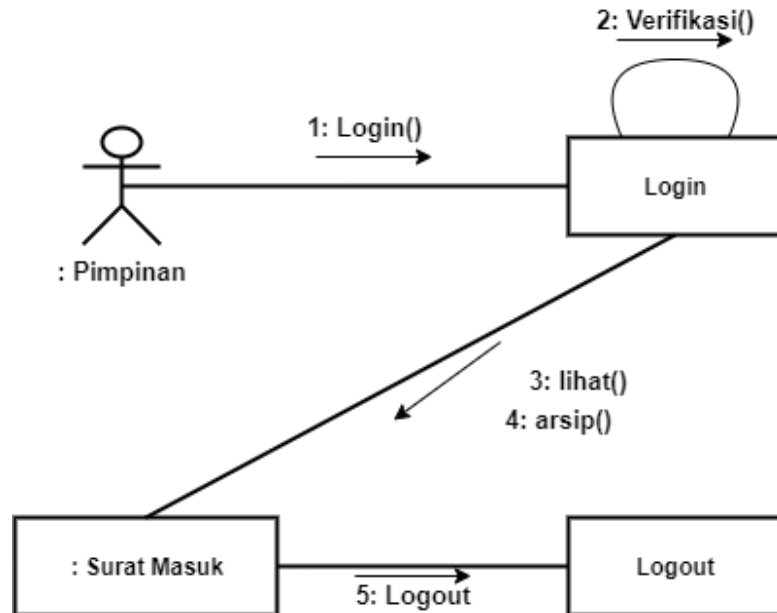
Gambar 3. 10 Collaboration Diagram Disposisi Pimpinan

c) Collaboration diagram Tracking Disposisi



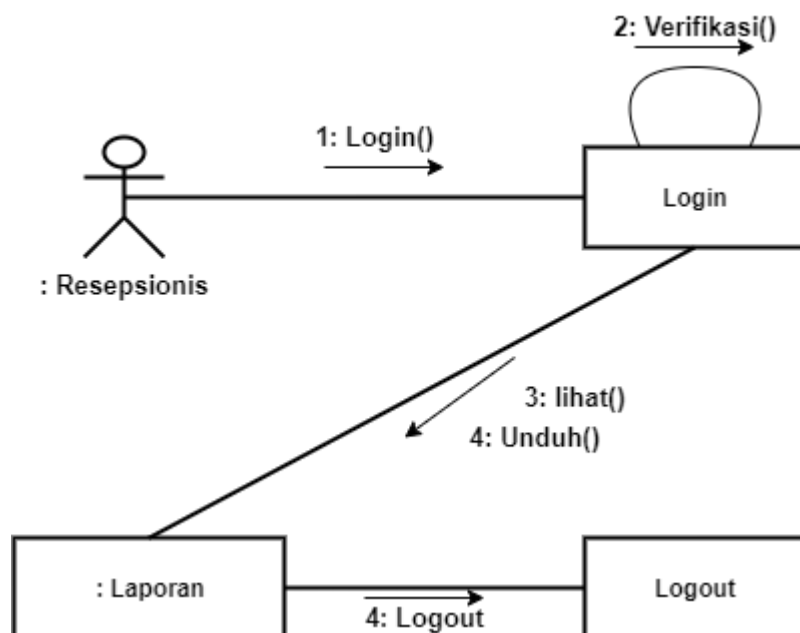
Gambar 3. 11 Collaboration Diagram Tracking Disposisi

d) *Collaboration diagram* Pengarsipan Surat Pimpinan



Gambar 3. 12 *Collaboration Diagram* Pengarsipan Surat Pimpinan

e) *Collaboration diagram* Unduh Laporan

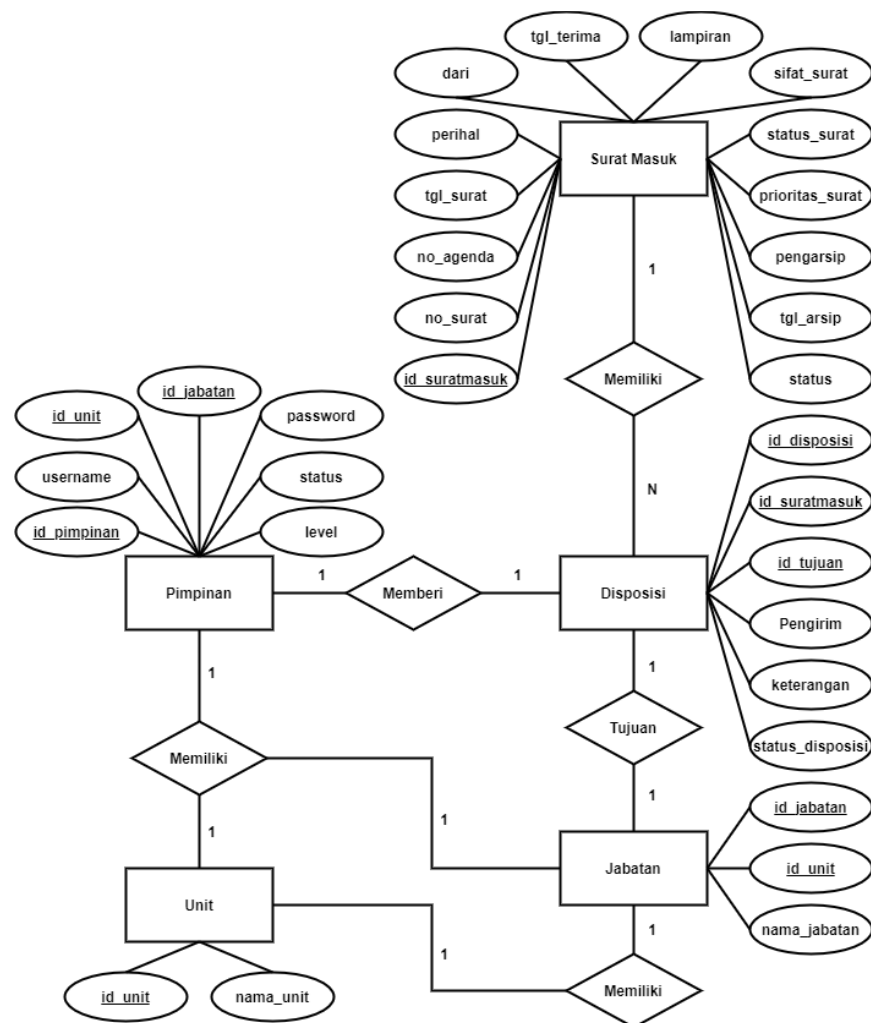


Gambar 3. 13 *Collaboration Diagram* Unduh Laporan

b. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan hubungan penterjemahan yang berisi komponen-komponen himpunan *entity* dan himpunan *relasi* yang dilengkapi dengan atribut-atribut menghubungkan *entity* tersebut digunakan *key field* (*Primary Key* Atribut) dari masing-masing *entity*.

Adapun bentuk *entity relationship diagram* dapat dilihat pada gambar dibawah ini :



Gambar 3. 14 *Entity Relationship Diagram* (ERD)

Pada Gambar 3.13 Surat Masuk memiliki banyak disposisi yang dihubungkan berdasarkan id_suratmasuk. Setiap Pimpinan akan menginput satu Disposisi, dan setiap disposisi memiliki satu tujuan

berdasarkan id_jabatan. Setiap Pimpinan memiliki satu id_jabatan dan id_unit, dan setiap Jabatan memiliki satu id_unit.

2. Perancangan Terinci

Desain sistem terinci merupakan lanjutan dari desain global yang menjelaskan tentang sistem secara lebih terinci. Desain secara terinci yang disebut juga dengan desain teknis sistem secara fisik atau disebut juga dengan desain internal, yaitu perancangan bentuk fisik atau bagan arsitektur sistem yang diusulkan.

Tujuan utama dari desain secara terinci ini adalah untuk memenuhi kebutuhan para pemakai sistem dan memberikan gambaran yang jelas dan rancang bangun yang lengkap yang nantinya digunakan untuk pembuatan program komputer, diperlukan bagi para ahli teknik dan bagian yang terlibat dalam pembangunan sistem informasi.

a. Desain Output

Desain output merupakan sebuah rancangan yang menggambarkan keluaran yang diinginkan dari sistem informasi yang ingin dirancang. Bentuk *output*/laporan yang dirancang untuk sistem informasi *E-Office* pada UIN Mahmud Yunus Batusangkar menggunakan metode *Model View Controller* (MVC) adalah sebagai berikut :

1) Laporan Disposisi

<div style="border: 1px solid black; width: 60px; height: 40px; margin: 0 auto; display: flex; align-items: center; justify-content: center;">Logo</div>	KEMENTERIAN AGAMA REPUBLIK INDONESIA UNIVERSITAS AGAMA ISLAM NEGERI MAHMUD YUNUS BATUSANGKAR
LEMBAR DISPOSISI	
Nomor surat : X(30)	Status : <input type="checkbox"/> Asli <input type="checkbox"/> Tebusan
Tanggal Surat : Mm/dd/yyyy	<input type="checkbox"/> Sangat Segera/Kilat <input type="checkbox"/> Segera <input type="checkbox"/> Biasa
Lampiran : X(30)	
Diterima Tanggal : Mm/dd/yyyy	
Nomor Agenda : X(30)	<input type="checkbox"/> Sangat Rahasia <input type="checkbox"/> Rahasia <input type="checkbox"/> Biasa
Dari : X(30)	
Perihal : X(30)	
Diteruskan kepada : 1. X(30) 2. X(30) 3. X(30)	
Disposisi Rektor Kepada :	Petunjuk :
<input type="checkbox"/> Warek I	<input type="checkbox"/> Setuju <input type="checkbox"/> Perbaiki
<input type="checkbox"/> Warek II	<input type="checkbox"/> Tolak <input type="checkbox"/> Bicarakan dg. Rektor
<input type="checkbox"/> Warek III	<input type="checkbox"/> Teliti <input type="checkbox"/> Bicarakan Bersama
<input type="checkbox"/> Kepala Biro AUAKA	<input type="checkbox"/> Untuk Diketahui <input type="checkbox"/> Ingatkan
<input type="checkbox"/> Direktur Pascasarjana	<input type="checkbox"/> Selesaikan <input type="checkbox"/> Simpan
<input type="checkbox"/> Dekan.....	<input type="checkbox"/> Sesuai Catatan <input type="checkbox"/> Disiapkan
<input type="checkbox"/> KA. LPM/LP2M *)	<input type="checkbox"/> Untuk Perhatian <input type="checkbox"/> Harap dihadiri/diwakili
	<input type="checkbox"/> Edarkan
	<input type="checkbox"/> Jawab
Catatan Rektor : X(30)	
Tanggal Penyelesaian : Mm/dd/yyyy	
Penerima : X(30)	

Gambar 3. 15 Laporan Disposisi

Pada Gambar 3.14 merupakan tampilan output dari disposisi surat masuk. Data disposisi dan surat masuk tersebut didapatkan dari *Model* yaitu Suratmasuk_model.php dan

Disposisi_model.php yang akan diterima oleh *Controller* dan kemudian *Controller* menampilkan data tersebut kepada pengguna melalui *View*.

2) Laporan Surat Masuk

Logo		KEMENTERIAN AGAMA REPUBLIKINDONESIA UNIVERSITAS AGAMA ISLAM NEGERI MAHMUD YUNUS BATUSANGKAR				
No	Nomor Surat	Dari	Tanggal Surat	Nomor Agenda	Lampiran	Perihal
X(10)	X(30)	X(30)	Mm/dd/yyyy	X(30)	X(30)	X(30)
X(10)	X(30)	X(30)	Mm/dd/yyyy	X(30)	X(30)	X(30)

Gambar 3. 16 Laporan Surat Masuk

3) Tracking Posisi Disposisi Surat

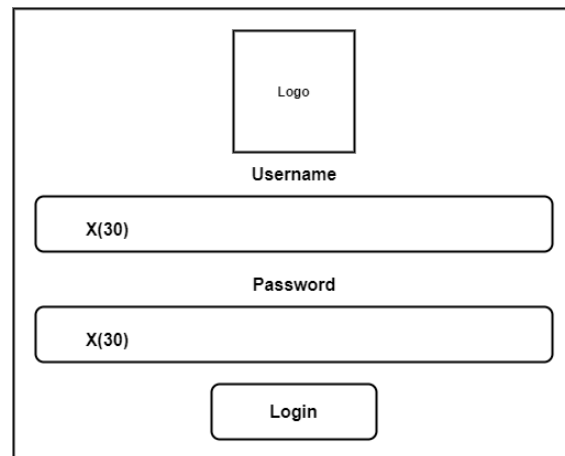
Tracking Posisi Disposisi Surat				
Diposisi Ke	Nomor Surat	Pengirim Disposisi	Penerima Disposisi	Isi Disposisi
X(10)	X(30)	X(30)	X(30)	X(30)
X(10)	X(30)	X(30)	X(30)	X(30)

Gambar 3. 17 Tampilan Tracking Disposisi Surat

b. Desain *Input*

Desain *input* diperlukan untuk memperoleh pengisian *database*, perlu dirancang *form-form input*, hal ini akan sangat membantu pada saat kasir memasukkan data. Bentuk *form/tampilan input* yang dirancang untuk sistem informasi *E-Office* pada UIN Mahmud Yunus Batusangkar menggunakan metode *Model View Controller* (MVC) sebagai berikut :

1) Tampilan *Login*



A diagram of a login form within a rectangular border. At the top center is a square box labeled "Logo". Below it is the label "Username". Under the label is a horizontal rectangular input field containing the text "X(30)". Below the input field is the label "Password". Under the label is another horizontal rectangular input field containing the text "X(30)". At the bottom center is a rounded rectangular button labeled "Login".

Gambar 3. 18 Tampilan *Login*

2) Tampilan *Input Surat Masuk*

Nomor Surat	ID Surat Masuk
<input type="text" value="X(30)"/>	<input type="text" value="9(11)"/>
Tanggal Surat	Status Surat
<input type="text" value="Mm/dd/yyyy"/>	<input type="text" value="X(30)"/> V
Lampiran	Prioritas Surat
<input type="text" value="X(30)"/>	<input type="text" value="X(30)"/> V
Tanggal Diterima	Sifat Surat
<input type="text" value="Mm/dd/yyyy"/>	<input type="text" value="X(30)"/> V
Nomor Agenda	File Surat
<input type="text" value="X(30)"/>	<input type="button" value="Pilih File"/> *pdf, jpg, png
Dari	
<input type="text" value="X(30)"/>	
Perihal	
<input type="text" value="X(30)"/>	
Pengarsip	Tanggal Arsip
<input type="text" value="X(30)"/>	<input type="text" value="Mm/dd/yyyy hh:mm:ss"/>
Status	Petunjuk
<input type="text" value="X(30)"/>	<input type="text" value="X(30)"/> V
<input type="button" value="Keluar"/>	<input type="button" value="Delete"/> <input type="button" value="Edit"/> <input type="button" value="Tambah"/>

Gambar 3. 19 Tampilan *Input* Surat Masuk

3) Tampilan *Input* Disposisi

ID Disposisi	Pengirim
9(11)	X(30)
ID Surat Masuk	Status Disposisi
9(11)	9(11) V
Unit	
X(30) V	
Jabatan	
X(30) V	
Isi Disposisi	
X(30)	
Disposisi	

Gambar 3. 20 Tampilan *Input Disposisi*

c. Desain File

Apabila desain *ouput* dan *input* selesai dirancang, langkah selanjutnya yang harus dilakukan adalah merancang *file-file database*. *File* ini berfungsi menyimpan dan mengelompokan data-data. Di dalam rancangan *file* ini dijelaskan variabel-variabel apa saja yang digunakan dalam *file-file* atau tabel-tabel yang terkait dalam *E-Office* UIN Mahmud Yunus Batusangkar menggunakan metode *Model View Controller* (MVC). Tabel-tabel tersebut sebagai berikut:

1) Tabel User

Nama *Database* : e-office

Nama Tabel : tb_user

Primary Key : id_user

Tabel 3. 1 User

<i>Field name</i>	<i>Type</i>	<i>Width</i>	<i>Description</i>
id_user	int	11	id user

username	varchar	125	nama user
password	varchar	125	<i>password</i>
id_unit	int	11	id unit
id_jabatan	int	11	id jabatan
status	int	11	status
level	int	11	level

2) Tabel Unit

Nama *Database* : e-office

Nama Tabel : tb_unit

Primary Key : id_unit

Tabel 3. 2 Unit

Field name	Type	Width	Description
id_unit	int	11	id unit
nama_unit	varchar	125	nama unit

3) Tabel Jabatan

Nama *Database* : e-office

Nama Tabel : tb_jabatan

Primary Key : id_jabatan

Tabel 3. 3 Jabatan

Field name	Type	Width	Description
id_jabatan	int	11	id jabatan
id_unit	int	11	id unit
nama_jabatan	varchar	125	nama jabatan

4) Tabel Surat Masuk

Nama *Database* : e-office

Nama Tabel : tb_surat_masuk

Primary Key : id_suratmasuk

Tabel 3. 4 Surat Masuk

Field name	Type	Width	Description
id_suratmasuk	int	11	id surat masuk
no_surat	varchar	125	nomor surat
tgl_surat	date	-	tanggal surat
lampiran	varchar	125	lampiran
tgl_terima	date	-	tanggal terima
no_agenda	varchar	125	nomor agenda
status_surat	varchar	125	status surat
prioritas_surat	varchar	125	prioritas surat
sifat_surat	varchar	125	sifat surat
dari	text	-	pengirim surat
perihal	text	-	perihal
petunjuk	varchar	125	petunjuk
pengarsip	varchar	125	pengarsip
tgl_arsip	datetime	-	tanggal arsip
status	int	11	status
file	text	-	file

5) Tabel Disposisi

Nama Database : e-office

Nama Tabel : tb_disposisi

Primary Key : id_disposisi

Tabel 3. 5 Disposisi

Field name	Type	Width	Description
id_disposisi	int	11	id disposisi
id_suratmasuk	int	11	id surat masuk
pengirim	varchar	125	pengirim
id_tujuan	int	11	id tujuan
keterangan	text	-	isi disposisi
status_disposisi	int	11	status disposisi

BAB IV

PENUTUP

A. Kesimpulan

E-Office adalah pekerjaan yang berhubungan dengan administrasi perkantoran seperti pengelolaan surat masuk, surat keluar dan disposisi surat yang dikerjakan menggunakan bantuan alat komunikasi dan sistem informasi sehingga dapat meningkatkan kinerja perkantoran, khususnya dalam pengolahan administrasi surat. Pembuatan sistem dalam penelitian ini menggunakan metode *Model View Controller* (MVC) yang membuat kode aplikasi menjadi lebih terstruktur. Pemisahan aplikasi menjadi 3 bagian, yaitu *Model*, *View* dan *Controller* sehingga kodingan menjadi lebih sederhana dan memudahkan dalam pengembangan sistem lanjutan.

Berdasarkan kesimpulan di atas maka penulis menyimpulkan bahwa dengan adanya sistem informasi *e-office* dengan menggunakan metode MVC ini maka dapat mempermudah dan memaksimalkan dalam proses administrasi surat masuk dan disposisi surat pada UIN Mahmud Yunus Batusangkar. Serta mempermudah dalam pembuatan laporan surat masuk dan laporan disposisi surat karna telah menggunakan database sehingga data surat tersimpan dengan aman dan baik.

B. Saran

Berdasarkan kesimpulan diatas, penulis memberikan saran agar Sistem Informasi *E-Office* pada UIN Mahmud Yunus Batusangkar bisa bermanfaat dan dikembangkan untuk penelitian selanjutnya karena masih ada proses administrasi surat yang belum terimplementasi seperti proses surat keluar dan agenda rapat ke dalam sistem ini.

DAFTAR PUSTAKA

- Abdul, K. (2014). *Pengenalan Sistem Informasi Edisi Revisi*. CV. Andi Offset.
- Ardhianto, E. (2015). Rancang Bangun E-Office Administrasi Agenda Kegiatan Promosi Unisbank Semarang. *Jurnal Teknologi Informasi DINAMIK*, 20(1), 17–24.
- Arief, M. R. (2011). *Pemrograman Web Dinamis Menggunakan PHP dan MYSQL*. Andi.
- Badiyanto. (2013). *Cara Mudah Membangun Aplikasi Web PHP*. Mediakom.
- Dewi, R. K., Adrian, Q. J., Sulistiani, H., & Isnaini, F. (2021). Dashboard Interaktif Untuk Sistem Informasi Keuangan Pada Pondok Pesantren Mazroatul'ulum. *Jurnal Teknologi Dan Sistem Informasi (JTISI)*, 2(2), 116–121. <http://jim.teknokrat.ac.id/index.php/JTISI>
- Hartono, B. (2013). *Sistem Informasi Manajemen Berbasis Komputer*. PT. Rineka Cipta.
- Jogiyanto. (2005). *Analisis & desain : sistem informasi : pendekatan terstruktur teori dan praktik aplikasi bisnis*. Andi Offset.
- Julianty, S., Nurhadi, & Rosario, M. (2019). Rancang Bangun Aplikasi E-Office Berbasis Web (Studi Kasus : Kantor kecamatan Sadu Kabupaten Tanjung Jabung Timur). *Jurnal Ilmiah Mahasiswa Teknik Informatika*, 1(3), 149–158.
- M. Shalahuddin, R. A. S. (2013). *Rekayasa Perangkat Lunak: Terstruktur dan Berorientasi Objek*. Penerbit Informatika.
- Maimunah, M., Sunarya, L., & Larasati, N. (2012). Media Company Profile Sebagai Sarana Penunjang Informasi Dan Promosi. *CCIT Journal*, 5(3), 281–301. <https://doi.org/10.33050/ccit.v5i3.155>
- Pasaribu, J. S., Informatika, T., Ganesha, P. P., Peserta, P., Baru, D., Web, B., & Based, W. (2017). *PENERAPAN FRAMEWORK YII PADA PEMBANGUNAN*. III(2), 154–163.
- Rahmadiansyah, Dudi & Irwan, D. (2012). Implementasi Metode Model View Controller Menggunakan Framework Code Igniter dalam Pengembangan Aplikasi Manajemen Depo Petikemas pada Unit Usaha Belawan Logistics Center. *Seminar Nasional Teknologi Informasi Dan Komunikasi (SNASTIKOM 2012)*, ISBN 978-6.

- Ramdhan, N. A., & Nufriana, D. A. (2019). Rancang Bangun Dan Implementasi Sistem Informasi Skripsi Oline Berbasis WEB. *Jurnal Ilmiah Intech : Information Technology Journal of UMUS*, 1(02), 1–12.
<https://doi.org/10.46772/intech.v1i02.75>
- Sapto Aji, Migunani, F. N. H. (2014). *Rancang Bangun Sistem Informasi Disposisi Surat Berbasis Web (Studi Kasus Kementerian Pekerjaan Umum)*. 3(3), 25–32.
- Setiadi, A., & Alfiah, F. (2016). Sistem Penjualan Spare Part Toko Ajm Motor Menggunakan Ci Berbasis Arsitektur Mvc. *Simetris : Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer*, 7(2), 575.
<https://doi.org/10.24176/simet.v7i2.770>
- Sudradjat, A., Atmaja, R. K., Ramadan, R., & Ispandi. (2022). Penerapan Konsep MVC Pada Perancangan Aplikasi Sistem Informasi Persediaan Alat Kebersihan. *Jurnal Infortech*, 4(1), 64–71.
<http://ejournal.bsi.ac.id/ejurnal/index.php/infortech64>
- Sutabri, T. (2012). *Konsep sistem informasi*. CV. Andi Offset.
- Syafitri, S. A., Pratama, A., & Ulva, A. F. (2020). Sistem Informasi Administrasi Persuratan (Paperless Office) Berbasis Web Pada Fakultas Teknik Universitas Malikussaleh. *Jurnal Ilmiah Sistem Informasi*, 4(1), 95–110.
<https://ojs.unimal.ac.id/sisfo/article/view/6278/3119>
- Taufiq, R. (2003). *Sistem Informasi Manajemen*. Graha Ilmu.
- Yakub. (2012). *Pengantar Sistem Informasi*. Graha Ilmu.