



**RANCANG BANGUN SISTEM PENJUALAN OBAT NON
GENERIK MENGGUNAKAN METODE *RAPID*
APPLICATION DEVELOPMENT (STUDI KASUS
APOTEK SETIA SEHAT MARELAN)**

Disusun dan Diajukan untuk Memenuhi Persyaratan Ujian Akhir Memperoleh
Gelar Sarjana Komputer pada Fakultas Sains dan Teknologi
Universitas Pembangunan Panca Budi
Medan

SKRIPSI

OLEH:

**NAMA : DEVI YULIANTI
NPM : 1714370189
PROGRAM STUDI : SISTEM KOMPUTER**

**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2021**

PENGESAHAN SKRIPSI

JUDUL : RANCANG BANGUN SISTEM PENJUALAN OBAT NON GENERIK
MENGUNAKAN METODE RAPID APPLICATION DEVELOPMENT (STUDI
KASUS APOTEK SETIA SEHAT MARELAN)

NAMA : DEVI YULIANTI
N.P.M : 1714370189
FAKULTAS : SAINS & TEKNOLOGI
PROGRAM STUDI : Sistem Komputer
TANGGAL KELULUSAN : 03 November 2021

DIKETAHUI

DEKAN



Hamdani, ST., MT.

KETUA PROGRAM STUDI



Eko Hariyanto, S.Kom., M.Kom

**DISETUJUI
KOMISI PEMBIMBING**

PEMBIMBING I



Arpan, S.Kom., M.Kom

PEMBIMBING II



Jodi Hendrawan, S.Kom., M.Kom

Hal : Permohonan Meja Hijau

FM-BPAA-2012-041

Medan, 13 Oktober 2021
 Kepada Yth : Bapak/Ibu Dekan
 Fakultas SAINS & TEKNOLOGI
 UNPAB Medan
 Di -
 Tempat

Dengan hormat, saya yang bertanda tangan di bawah ini :

Nama : DEVI YULIANTI
 Tempat/Tgl. Lahir : BULU CINA / 09/11/1999
 Nama Orang Tua : JUMADI
 N. P. M : 1714370189
 Fakultas : SAINS & TEKNOLOGI
 Program Studi : Sistem Komputer
 No. HP : 085261642281
 Alamat : Jl. KLY. SUDARSO NO. 5B LK III KOTA BANGUN

Datang bermohon kepada Bapak/Ibu untuk dapat diterima mengikuti Ujian Meja Hijau dengan judul Rancang Bangun Sistem Penjualan Obat Non Generik Menggunakan Metode Rapid Application Development (Studi Kasus Apotek Setia Sehat Marelan), Selanjutnya saya menyatakan :

1. Melampirkan KKM yang telah disahkan oleh Ka. Prodi dan Dekan
2. Tidak akan menuntut ujian perbaikan nilai mata kuliah untuk perbaikan Indeks prestasi (IP), dan mohon diterbitkan Ijazahnya setelah lulus ujian meja hijau.
3. Telah tercap keterangan bebas pustaka
4. Terlampir surat keterangan bebas laboratorium
5. Terlampir pas photo untuk Ijazah ukuran 4x6 = 5 lembar dan 3x4 = 5 lembar Hitam Putih
6. Terlampir foto copy STTB SLTA dilegalisir 1 (satu) lembar dan bagi mahasiswa yang lanjutan D3 ke S1 lampirkan ijazah dan transkripnya sebanyak 1 lembar.
7. Terlampir pelunasan kwintasi pembayaran uang kuliah berjalan dan wisuda sebanyak 1 lembar
8. Skripsi sudah dijilid lux 2 exemplar (1 untuk perpustakaan, 1 untuk mahasiswa) dan jilid kertas jeruk 5 exemplar untuk penguji (bentuk dan warna penjiilidan diserahkan berdasarkan ketentuan fakultas yang berlaku) dan lembar persetujuan sudah di tandatangani dosen pembimbing, prodi dan dekan
9. Soft Copy Skripsi disimpan di CD sebanyak 2 disc (Sesuai dengan Judul Skripsinya)
10. Terlampir surat keterangan BKKOL (pada saat pengambilan ijazah)
11. Setelah menyelesaikan persyaratan point-point diatas berkas di masukan kedalam MAP
12. Bersedia melunaskan biaya-biaya uang dibebankan untuk memproses pelaksanaan ujian dimaksud, dengan rincian sbb :

1. [102] Ujian Meja Hijau	: Rp.	1,000,000
2. [170] Administrasi Wisuda	: Rp.	1,750,000
Total Biaya	: Rp.	2,750,000

Ukuran Toga :

M

Diketahui/Disetujui oleh :



Hamdani, ST., MT.
 Dekan Fakultas SAINS & TEKNOLOGI

Hormat saya



DEVI YULIANTI
 1714370189

Catatan :

- 1. Surat permohonan ini sah dan berlaku bila ;
 - o a. Telah dicap Bukti Pelunasan dari UPT Perpustakaan UNPAB Medan.
 - o b. Melampirkan Bukti Pembayaran Uang Kuliah aktif semester berjalan.
- 2. Dibuat Rangkap 3 (tiga), untuk - Fakultas - untuk BPAA (asti) - Mhs.ybs.

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Devi Yulianti

NPM : 1714370189

Prodi : Sistem Komputer

JudulSkripsi : RANCANG BANGUN SISTEM PENJUALAN OBAT
NON GENERIK MENGGUNAKAN METODE *RAPID
APPLICATION DEVELOPMENT*

Dengan ini menyatakan bahwa :

1. Skripsi ini merupakan hasil karya tulis saya sendiri dan bukan merupakan hasil karya tulis oranglain.
2. Memberi izin hak bebas Royalti Non-Eksklusif kepada UNPAB untuk menyimpan, mengalih-media/ formatkan pengelola mendistribusikan, dan mempublikasikan karya skripsinya melalui internet dan media lain bagi kepentingan akademik.

Pernyataan ini saya perbuat dengan penuh tanggung jawab dan saya bersedia menerima konsekuensi apapun sesuai dengan aturan yang berlaku apabila dikemudian hari diketahui apabila pernyataan ini tidak benar.

Medan, November 2021



(Devi Yulianti)

SURAT PERNYATAAN ORISINALITAS

Dengan ini menyatakan bahwa dalam skripsi ini tidak terdapat karya yang diajukan untuk memperoleh gelar kesarjanaan didalam perguruan tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis di dalam skripsi ini dan disebutkan dalam daftar pustaka.

Medan, November 2021



Devi Yulianti

NPM : 1714370189



UNIVERSITAS PEMBANGUNAN PANCA BUDI FAKULTAS SAINS & TEKNOLOGI

Jl. Jend. Gatot Subroto Km 4,5 Medan Fax. 061-8458077 PO.BOX : 1099 MEDAN

PROGRAM STUDI TEKNIK ELEKTRO	(TERAKREDITASI)
PROGRAM STUDI ARSITEKTUR	(TERAKREDITASI)
PROGRAM STUDI SISTEM KOMPUTER	(TERAKREDITASI)
PROGRAM STUDI TEKNIK KOMPUTER	(TERAKREDITASI)
PROGRAM STUDI AGROTEKNOLOGI	(TERAKREDITASI)
PROGRAM STUDI PETERNAKAN	(TERAKREDITASI)
PROGRAM STUDI TEKNOLOGI INFORMASI	(TERAKREDITASI)

PERMOHONAN JUDUL TESIS / SKRIPSI / TUGAS AKHIR*

Saya yang bertanda tangan di bawah ini :

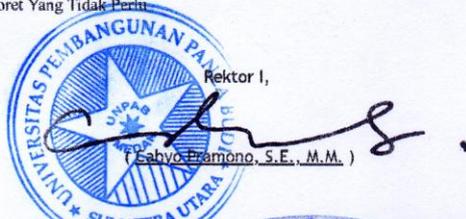
Nama Lengkap : DEVI YULIANTI
 Tempat / Tgl. Lahir : Bulu Cina / 09 November 1999
 Nomor Pokok Mahasiswa : 1714370189
 Program Studi : Sistem Komputer
 Konsentrasi : Keamanan Jaringan Komputer
 Jumlah Kredit yang telah dicapai : 143 SKS, IPK 3.83
 Nomor Hp : 085261642281

Dengan ini mengajukan judul sesuai bidang ilmu sebagai berikut :

No.	Judul
1.	Rancang Bangun Sistem Penjualan Obat Non Generik Menggunakan Metode Rapid Application Development (Studi Kasus Apotek Setia Sehat Marelan)

Catatan : Diisi Oleh Dosen Jika Ada Perubahan Judul

*Coret Yang Tidak Perlu



Medan, 13 November 2021

Pemohon,

(Devi Yulianti)

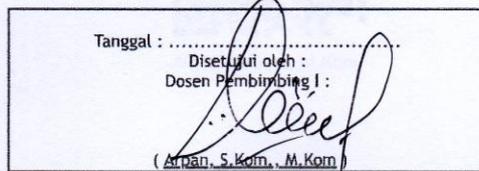


Tanggal :

Disetujui oleh :

Dekan I :

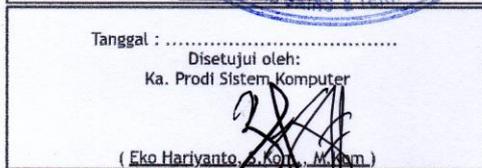
(Haidani, S.Ts., M.T.)



Tanggal :

Disetujui oleh :
Dosen Pembimbing I :

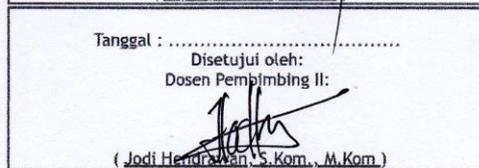
(Ardan, S.Kom., M.Kom.)



Tanggal :

Disetujui oleh:
Ka. Prodi Sistem Komputer

(Eko Hariyanto, S.Kom., M.Kom.)



Tanggal :

Disetujui oleh:
Dosen Pembimbing II :

(Jodi Hendriawan, S.Kom., M.Kom.)

No. Dokumen: FM-UPBM-18-02

Revisi: 0

Tgl. Eff: 22 Oktober 2018

Sumber dokumen: <http://mahasiswa.pancabudi.ac.id>

Dicetak pada: Sabtu, 13 November 2021 09:22:42

ABSTRAK

DEVI YULIANTI

**Rancang Bangun Sistem Penjualan Obat Non Generik Menggunakan
Metode *Rapid Application Development* (Studi Kasus Apotek Setia
Sehat Marelan)
2021**

Apotek merupakan tempat dimana konsumen dapat membeli obat generik dan non generik termasuk Apotek Setia Sehat Marelan. Apotek ini adalah apotek dimana penulis melakukan penelitian. Penjualan obat paten yang diteliti pada penelitian ini akan menghasilkan sistem informasi penjualan obat non generik yang dapat digunakan oleh pihak apotek. Sistem informasi yang dibangun dengan menggunakan metode *Rapid Application Development* dapat memberikan kemudahan kepada pihak apotek dalam melakukan pencatatan penjualan obat tersebut. Pencatatan secara manual dapat dihindarkan dengan adanya sistem informasi ini. Penggunaan metode *RAD* mengembangkan prototype sistem informasi dengan melakukan pendataan *user experience* dari penggunaan sistem informasi. Percobaan berdasarkan *user experience* dapat memberikan *feedback* sehingga dapat dikembangkan lebih lanjut. Sistem informasi penjualan obat non generik dapat menjadi sistem informasi yang lebih baik.

Kata Kunci: apotek, sistem informasi, paten, non generik

KATA PENGANTAR

Puji syukur ke hadirat Allah SWT, karena dengan berkat dan kasih anugerah-Nya penulis masih diberikan kesempatan untuk menyelesaikan skripsi ini sebagaimana mestinya. Skripsi ini berjudul "**Rancang Bangun Sistem Penjualan Obat Non Generik Menggunakan Metode *Rapid Application Development* (Studi Kasus Apotek Setia Sehat Marelan)**". Dalam kesempatan ini, penulis mengucapkan rasa terima kasih yang tak terhingga kepada pihak-pihak yang telah membantu dalam penyelesaian skripsi ini. Penulis ingin mengucapkan terima kasih kepada :

1. Orang tua saya yang selalu memberikan semangat, dukungan dan motivasi dalam penyusunan skripsi ini.
2. Bapak Dr. H. Muhammad Isa Indrawan, S.E., M.M., selaku Rektor Universitas Pembangunan Panca Budi Medan.
3. Bapak Hamdani, S.T., M.T., selaku Dekan Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan.
4. Bapak Eko Hariyanto, S.Kom., M.Kom., selaku Ketua Program Studi Sistem Komputer Universitas Pembangunan Panca Budi Medan.
5. Bapak Arpan, S.Kom., M.Kom., selaku Dosen Pembimbing I yang telah memberikan arahan dan membimbing dalam penyelesaian skripsi ini.
6. Bapak Jodi Hendrawan, S.Kom., M.Kom., selaku Dosen Pembimbing II yang telah memberikan ilmu pengetahuan, serta bimbingan dalam penyelesaian skripsi ini.
7. Dosen-dosen pada Program Studi Sistem Komputer Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan.
8. Staff dan karyawan Universitas Pembangunan Panca Budi Medan.
9. Muhammad Anil Rasyid, Sri Ayu Rahmadani dan Rekan-rekan dari program studi Sistem Komputer, Fakultas Sains dan Teknologi, Universitas Pembangunan Panca Budi, Medan.

Penulis juga menyadari bahwa penyusunan skripsi ini belum sempurna baik dalam penulisan maupun isi disebabkan keterbatasan kemampuan penulis. Oleh karena itu, penulis mengharapkan kritik dan saran yang sifatnya membangun dari pembaca untuk kesempurnaan isi skripsi ini.

Medan, 13 Agustus 2021
Penulis

Devi Yulianti
1714370189

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
DAFTAR GAMBAR	v
DAFTAR TABEL	vi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
BAB II LANDASAN TEORI	5
2.1 Sistem.....	5
2.1.1 Unsur-Unsur Sistem	6
2.1.2 Elemen Sistem	7
2.1.3 Pengelompokan Sistem	8
2.1.4 Pembagian Sistem	9
2.1.5 Contoh Sistem	11
2.2 Metode <i>Rapid Application Development</i>	12
2.2.1 Definisi	12
2.2.2 Manfaat.....	12
2.2.3 Tahapan	13
2.2.4 Prototipe Pembuatan Prototipe.....	15
2.2.5 Keterlibatan Pengguna	15
2.2.6 Kelayakan	15
2.2.7 Pengurangan Kesalahan & Debugging.....	16
2.2.8 Kelebihan Pengembangan Aplikasi yang Cepat	16
2.2.9 Kemajuan yang Dapat Diukur	16
2.2.10 Menghasilkan Kode Produktif dengan Cepat.....	16
2.2.11 Kompartementalisasi Komponen Sistem	17
2.2.12 Umpan Balik Pengguna yang Cepat dan Konstan:.....	17
2.2.13 Integrasi Sistem Awal	17
2.2.14 Adaptabilitas Sederhana	18
2.2.15 Kontra Pengembangan Aplikasi Cepat.....	18
2.2.16 Memerlukan Sistem Modular	18
2.2.17 Kesulitan Dalam Proyek Skala Besar.....	18
2.2.18 Menuntut Antarmuka Pengguna yang Sering	19
2.3 Apotek Setia Sehat Marelan.....	19
2.4 <i>Unified Modeling Language</i>	19
2.4.1 <i>Use Case Diagram</i>	20
2.4.2 <i>Activity Diagram</i>	21
2.4.3 <i>Sequence Diagram</i>	23

2.5	Diagram Alir	24
2.6	Pemrograman	27
2.6.1	Bahasa Pemrograman	27
2.6.2	Pemrograman <i>Web</i>	28
2.6.3	Desain <i>Web</i>	29
2.6.4	PHP	29
2.6.5	Cascading Style Sheets	30
2.7	<i>Database</i>	32
2.7.1	<i>Structured Query Language</i>	32
2.7.2	Evolusi <i>Database</i>	33
2.8	Tipe Data	33
2.8.1	Tipe Data <i>Numeric</i>	34
2.8.2	Tipe Data <i>String</i>	35
2.8.3	Tipe Data <i>Date</i>	36
2.9	<i>Visual Studio Code</i>	36
2.10	<i>World Wide Web</i>	38
2.10.1	Definisi <i>Web</i>	38
2.10.2	Server <i>Web</i>	40
2.10.3	Situs <i>Web</i>	41
BAB III METODE PENELITIAN		42
3.1	Tahapan Penelitian	42
3.2	Metode Pengumpulan Data	44
3.3	Pemodelan Penelitian	45
3.3.1	<i>Use Case Diagram</i>	45
3.3.2	<i>Activity Diagram</i>	46
3.3.3	<i>Sequence Diagram</i>	48
3.3.4	<i>Flowchart</i>	48
3.4	Perancangan <i>Interface</i>	50
3.4.1	Rancangan Beranda	50
3.4.2	Rancangan Obat Non Generik	51
3.4.3	Rancangan Penjualan	52
3.4.4	Rancangan Login	52
3.4.5	Rancangan Admin	53
3.4.6	Rancangan Pembeli	54
3.4.7	Rancangan Jenis Obat	55
3.5	Data Obat Non Generik	56
BAB IV HASIL DAN PEMBAHASAN		58
4.1	Spesifikasi Perangkat Keras	58
4.2	Spesifikasi Perangkat Lunak	59
4.3	Antarmuka Website	59
4.3.1	Halaman Beranda	59
4.3.2	Halaman Obat Non Generik	60
4.3.3	Halaman Jenis Obat	61
4.3.4	Halaman Pembeli	62

4.3.5	Halaman Penjualan.....	63
4.3.6	Halaman Laporan Penjualan.....	66
4.3.7	Halaman Admin	67
4.3.8	Halaman Login	68
4.3.9	Halaman Obat.....	69
4.3.10	Halaman Cek Pemesanan	70
BAB V PENUTUP.....		71
5.1	Kesimpulan	71
5.2	Saran.....	71

DAFTAR PUSTAKA

DAFTAR GAMBAR

Gambar 2.1 <i>Editor Visual Studio Code</i>	38
Gambar 3.1 Tahapan Penelitian	42
Gambar 3.2 <i>Use case diagram</i> penjualan obat.....	46
Gambar 3.3 <i>Activity diagram</i> penjualan obat.....	47
Gambar 3.4 <i>Sequence diagram</i> penjualan.....	48
Gambar 3.5 <i>Flowchart</i> penjualan obat.....	49
Gambar 3.6 Rancangan beranda	50
Gambar 3.7 Rancangan obat non generik	51
Gambar 3.8 Rancangan penjualan.....	52
Gambar 3.9 Rancangan login	53
Gambar 3.10 Rancangan admin	54
Gambar 3.11 Rancangan pembeli	55
Gambar 3.12 Rancangan jenis obat.....	56
Gambar 4.1 Halaman beranda.....	60
Gambar 4.2 Halaman obat non generik.....	61
Gambar 4.3 Halaman jenis obat	62
Gambar 4.4 Halaman pembeli.....	63
Gambar 4.5 Halaman penjualan	64
Gambar 4.6 Halaman detail penjualan	65
Gambar 4.7 Halaman cetak faktur	66
Gambar 4.8 Halaman laporan penjualan	67
Gambar 4.9 Halaman admin	68
Gambar 4.10 Halaman login	69
Gambar 4.11 Halaman obat.....	69
Gambar 4.12 Halaman cek pemesanan	70

DAFTAR TABEL

Tabel 2.1 Contoh Sistem	11
Tabel 2.2 Elemen <i>Use Case Diagram</i>	20
Tabel 2.3 Elemen <i>Activity Diagram</i>	22
Tabel 2.4 Elemen <i>Sequence Diagram</i>	23
Tabel 2.5 Simbol <i>Flowchart</i>	26
Tabel 2.6 Tipe Data <i>Numeric</i>	35
Tabel 2.7 Tipe Data <i>String</i>	35
Tabel 2.8 Tipe Data <i>Date</i>	36
Tabel 3.1 Data produk obat non generik.....	56
Tabel 4.1 Spesifikasi Perangkat Keras.....	58
Tabel 4.2 Spesifikasi Perangkat Lunak.....	59

BAB I

PENDAHULUAN

1.1 Latar Belakang

Obat-obatan adalah hal yang paling penting untuk mengobati pasien atau orang yang sedang mengalami sakit pada tubuhnya, termasuk sakit ringan dan sakit berat. Dalam memilih obat, tidak semua orang dapat mengetahui fungsi dan kegunaan obat tersebut. Ada obat yang bertipe generik dan ada juga obat yang bertipe paten. Obat generik memiliki harga ekonomis sementara obat paten memiliki harga yang jauh lebih mahal. Obat paten atau obat non generik dianggap lebih ampuh dalam menyembuhkan penyakit terutama penyakit yang serius.

Pada saat ini setiap apotek memiliki obat generik dan obat paten. Penelitian ini hanya mengambil obat paten untuk membangun sistem informasi penjualan obat tersebut. Apotek Setia Sehat Marelan merupakan salah satu apotek dimana penulis melakukan penelitian. Penulis juga bekerja di apotek tersebut. Pada saat ini penjualan hanya didata menggunakan Microsoft Excel sehingga dalam proses penjumlahan pembelian dan penjualan, pencatat stok dan perhitungan lainnya sering mengalami kendala. Hal ini akan mengakibatkan tidak sinkron data komputer dengan data sebenarnya.

Penulis tertarik untuk membangun sistem informasi yang dapat melakukan pencatatan stok dan penjualan terhadap obat non generik sehingga dapat memudahkan pihak apotek dalam melakukan pengawasan terhadap kualitas dari obat tersebut. Penulis ingin membuat sistem informasi penjualan obat non generik

yang memiliki informasi terhadap obat tersebut dengan jelas sehingga pada saat obat tersebut diberikan ke pelanggan, tidak ada kesalahan yang dapat timbulkan. Selain itu sistem informasi ini dapat diakses oleh pengguna dalam melakukan pembelian obat di apotek tersebut. Pelanggan dapat langsung membeli dari *website* sehingga penjualan langsung masuk ke sistem.

Metode *Rapid Application Development (RAD)* adalah metode yang berfokus pada pengembangan aplikasi secara cepat, melalui pengulangan dan *feedback* berulang-ulang (Aliya, 2021). Dalam membangun sistem informasi penjualan obat generik, metode ini berperan untuk menentukan alur dan proses sistem informasi tersebut. Percobaan berdasarkan *user experience* dilakukan untuk mendapatkan *feedback* dari pengguna sistem tersebut dalam proses pengembangan lebih lanjut sehingga sistem informasi penjualan obat non generik dapat dikembangkan dan berkembang menjadi sistem informasi yang lebih baik kedepannya.

Sistem informasi ini akan memberikan informasi terhadap harga, masa pakai obat yang sudah tersedia di apotek dan beberapa deskripsi pengetahuan tentang masing-masing obat yang dijual di apotek tersebut. Dengan adanya sistem ini pihak apotek dapat memantau kualitas obat non generik yang dimiliki oleh apotek tersebut. Berdasarkan latar belakang yang dipaparkan, penulis tertarik mengambil judul “**Rancang Bangun Sistem Penjualan Obat Non Generik Menggunakan Metode *Rapid Application Development* (Studi Kasus Apotek Setia Sehat Marelan)**”.

1.2 Rumusan Masalah

Berikut ini beberapa rumusan masalah yang ditetapkan dalam penelitian ini, yaitu:

1. Bagaimana membangun sistem informasi penjualan obat non generik di Apotek Setia Sehat Marelan?
2. Bagaimana menerapkan metode *Rapid Application Development* dalam sistem informasi penjualan yang akan dirancang?

1.3 Batasan Masalah

Berikut ini beberapa batasan masalah yang ditetapkan dalam penelitian ini, yaitu:

1. Obat yang digunakan sudah berstandar nasional Indonesia.
2. Program dirancang menggunakan *PHP* dan *MySQL*.
3. Panel admin diperuntukkan untuk pegawai apotek yang memiliki hak akses admin.
4. Sistem informasi hanya membahas mengenai penjualan obat non generik.
5. Sistem informasi dapat digunakan pengguna dalam melakukan pembelian.
6. *Web Server* yang digunakan adalah *Apache*.
7. Metode *RAD* hanya membahas pengelolaan data apotik dengan sekali percobaan.

1.4 Tujuan Penelitian

Berikut ini beberapa tujuan penelitian yang ditetapkan dalam penelitian ini, yaitu:

1. Untuk membangun sistem informasi penjualan obat non generik di Apotek Setia Sehat Marelan.
2. Untuk menerapkan metode *Rapid Application Development* dalam sistem informasi penjualan yang akan dirancang.

1.5 Manfaat Penelitian

Berikut ini beberapa manfaat penelitian yang ditetapkan dalam penelitian ini, yaitu:

1. Memberi informasi obat non generik yang akan dijual.
2. Membantu pihak apotek dalam melakukan penjualan obat tersebut.
3. Memberikan kontribusi ilmu kepada universitas dan penulis terhadap metode *Rapid Application Development*.

BAB II

LANDASAN TEORI

2.1 Sistem

Dalam asal etimologisnya, sistem ini berasal dari bahasa Yunani *Sustema* atau Latin *Systema*. Berbeda dengan banyak komponen lain (misalnya, integral), sistem adalah sesuatu yang dibentuk dari kombinasi banyak bagian (misalnya, *arbeitskreise*) dan bekerja sama menjadi satu unit (misalnya, *werkstatt*). Untuk mencegah hal itu terjadi, kami ingin setiap sistem menyertakan beberapa elemen, seperti beberapa bagian yang membentuk sistem utama (Fatta, 2017).

Istilah sistem digunakan dalam banyak aspek kehidupan kita. Sistem terdiri dari bagian-bagian yang masing-masing memiliki tugas berbeda yang saling melengkapi. Unsur-unsur tersebut harus membentuk hubungan satu sama lain dengan cara tertentu untuk menghasilkan tanggapan tertentu. Ini akan memungkinkan untuk memenuhi tujuan tertentu tepat waktu. Jika ada salah satu komponen dalam sistem ini yang tidak berfungsi dengan benar, maka akan menyebabkan seluruh sistem rusak (Jogiyanto, 2016).

Hal inilah yang membuat organisasi menjadi sangat penting; kerjasama dan integrasi antar bagiannya. Sebagai salah satu bagian dari alur kerja, suatu sistem menggabungkan berbagai unsur yang menyusun cara kerja sistem tersebut. Sebagai contoh, menemukan sesuatu gangguan di proses sistem tersebut, dampaknya akan langsung berakibat pada proses lain juga (Ladjamudin, 2017).

2.1.1 Unsur-Unsur Sistem

Selain berbagai aspek, sistem yang kompleks dibentuk dari banyak elemen dan komponen. Hal inilah yang kami harapkan akan tercipta sistem yang lengkap. Elemen resmi dari sistem termasuk.

1. Bahan dan Komponen

Komponen pertama dari produk makanan adalah komponen yang membuat makanan itu seperti apa adanya. Akan ada usus, hati, pankreas, dan semua organ internal lainnya di sistem pencernaan. Dengan semua elemen ini, sistem yang luar biasa dapat dibentuk.

2. Hubungan dan sinergi

Semua bagian dari sistem harus mencapai tujuan yang dimaksudkan karena mereka cocok untuk tugas yang berbeda dan kekuatan satu sama lain. Jika tidak terkoneksi dengan baik maka proses kerja akan berjalan tidak sempurna.

3. Lingkungan dan lokasi

Jika material secara keseluruhan sudah memiliki konstituen, maka diperlukan juga wadah. Meskipun hal ini bervariasi pada sistem, Anda berencana untuk membangunnya — sistem operasi dan semua teknologi yang terkait.

4. Integrasi

Sangat erat kaitannya dengan yang sebelumnya, yaitu sifat unsur-unsurnya. Semuanya harus dilakukan sesuai dengan yang direncanakan untuk memaksimalkan efisiensi.

5. Tujuan

Mengapa menempatkan pekerjaan ke dalam sistem jika tidak akan ada gunanya. Tujuan utama dari sistem ini adalah untuk menyederhanakan proses. Salah satu contohnya adalah sistem yang menjaga pertumbuhan ekonomi atau “mencoba untuk menjaga agar roda perekonomian tetap berputar”.

2.1.2 Elemen Sistem

Elemen sistem adalah bagian diskrit dari sistem yang dapat diimplementasikan untuk memenuhi persyaratan yang ditentukan. Elemen sistem dapat berupa perangkat keras, perangkat lunak, data, manusia, proses (misalnya, proses untuk memberikan layanan kepada pengguna), prosedur (misalnya, instruksi operator), fasilitas, bahan, dan entitas yang terjadi secara alami (misalnya, air, organisme, mineral), atau kombinasi apa pun (Hutahaean, 2015).

Ada tujuh bagian elemen sistem, antara lain:

6. Tujuan

Sesuatu yang menjadi titik akhir dan harapan akan hasil yang dicapai. Hal ini bisa menjadi acuan untuk membentuk sistem yang baik agar hasilnya sesuai.

7. *Input*

Segala hal yang dimasukkan untuk diproses baik itu berupa unsur fisik maupun non fisik. Tanpa adanya input, tidak akan ada sesuatu yang bisa diproses.

8. Proses

Urutan untuk mengubah unsur input menjadi hasil *output* melalui komponen yang saling berkaitan satu sama lain.

9. *Output*

Output adalah hasil yang muncul setelah unsur *input* selesai diproses. Hal ini bisa berupa objek fisik atau abstrak berupa data/informasi.

10. Batas

Pemisah antara sistem dan lingkungan luar sehingga tidak mengganggu satu sama lain.

11. Kontrol dan *Feedback*

Sebagai bentuk pengendalian terhadap sistem dengan menggunakan hasil output sebagai *feedback* untuk proses input selanjutnya.

12. Lingkungan

Lingkungan adalah ekosistem di sekitar yang mampu mempengaruhi baik secara positif ataupun negatif.

2.1.3 Pengelompokan Sistem

Sistem bisa dikelompokkan berdasarkan dua hal, dari segi komponen dan juga keterbukaannya. Adapun pengelompokan sistem dapat dilihat seperti berikut ini (Hutahaean, 2015):

1. Berdasarkan Komponen

- a. Sistem fisik adalah sistem yang terbuat dari benda-benda nyata yang ada dan dapat dilihat dengan mata telanjang kita. Misalnya sistem pencernaan, kita mengenal bagian utamanya.
- b. Sistem di mana unsur penyusunnya bukan fisik. Artinya tidak berwujud fisik karena hanya dalam konsep, gagasan, atau inovasi tertentu.

2. Berdasarkan Keterbukaan Sistem

- a. Sistem terbuka, jika suatu sistem saling terkait, dapat dimodifikasi dan dipengaruhi oleh objek luar.
- b. Sistem tertutup, di mana semuanya beroperasi di dalam sistem dan tidak bergantung pada elemen luar.

2.1.4 Pembagian Sistem

Sistem mengintegrasikan satu komponen dengan komponen lainnya karena dirancang untuk tujuan berbeda untuk setiap peristiwa. Seperti yang kita ketahui, sistem dapat diklasifikasikan dari sudut pandang yang berbeda. Sistem diklasifikasikan menggunakan berikut ini (Hutahaean, 2015):

1. Sistem Fisik dan Abstrak

Sistem pemikiran, seperti sistem teologi, adalah sistem yang bukan ide fisik. Sistem fisik diartikan sebagai sistem yang dapat dilihat secara fisik, seperti sistem komputer.

2. Sistem alami dan buatan

Sistem alam adalah hal-hal yang terjadi hanya melalui proses alam di alam, bukan konstruksi buatan manusia. Perangkat lunak buatan manusia adalah perangkat lunak yang dirancang manusia — sistem yang melibatkan interaksi manusia, seperti sistem keuangan, sistem manajemen, dan sistem informasi.

3. Sistem diskrit dan kontinu.

Sistem deterministik secara tradisional adalah sistem yang dapat diprediksi. Perilaku bagian-bagian sistem dapat ditentukan dengan tepat. Sehingga keluaran dari sistem dapat diprediksi. Misalnya, sistem komputer adalah contoh sistem yang keluarannya dapat diprediksi berdasarkan program yang sedang berjalan. Sedangkan sistem probabilistik tunduk pada elemen kebetulan, contohnya sistem manusia.

4. Sistem terbuka dan tertutup.

Sistem terbuka adalah sistem yang dipengaruhi oleh lingkungannya dalam berbagai cara. Ia juga dikenal sebagai sistem otomatis, yang merupakan bagian dari sistem buatan manusia dan berinteraksi dengan kontrol oleh komputer sebagai bagian dari sistem yang digunakan dalam masyarakat modern. Sistem ini merespon masukan dan digunakan sebagai keluaran oleh subsistem lain. Sistem tertutup adalah sistem yang tidak berinteraksi dengan lingkungan dan tidak terpengaruh olehnya. Sistem ini beroperasi secara otomatis terlepas dari apa yang dilakukan oleh sumber daya eksternal. Secara teoritis, sistem itu ada; Namun, dalam praktiknya, tidak

ada sistem yang terisolasi. Itu hanya diisolasi sampai tingkat tertentu (Omar Pahlevi et al., 2018).

2.1.5 Contoh Sistem

Ada beberapa contoh sistem yang dibagi menjadi beberapa bagian. Tabel 2.1 menunjukkan contoh beberapa sistem.

Tabel 2.1 Contoh Sistem

Jenis Sistem	Keterangan
Sistem ekonomi	Sistem ini mengatur semua aktivitas ekonomi, dan aturan serta prosedurnya mengatur semua transaksi yang ada. Tujuannya untuk meningkatkan perekonomian masyarakat dengan mendorong perkembangan usaha.
Sistem pemerintahan	Sistem pemerintahan terdiri dari banyak unsur, antara lain pemerintahan, legislatif, dan presiden. Berbagai elemen ini bekerja sama untuk menciptakan keadaan yang teratur.
Sistem Kendaraan	Sistem kendaraan adalah sistem di mana semua bagian kendaraan bersatu untuk terjadi dengan benar. Konstituennya, termasuk fluida kerja, mulai dengan busi, karburator, dan komponen sejenis lainnya.
Sistem Operasi	Perangkat lunak inilah yang membuat perangkat itu hidup. Sistem operasi adalah sistem non fisik karena mata tidak dapat melihatnya.

Sumber: (Omar Pahlevi et al., 2018)

2.2 Metode *Rapid Application Development*

Metode *Rapid Application Development (RAD)* ditemukan oleh James Martin pada tahun 1991. Meskipun sudah ada cukup lama, pendekatan *RAD* masih populer di antara mereka yang mencari metode tangkas pengembangan aplikasi untuk mengimbangi pertumbuhan bisnis dan klien. kebutuhan. *Rapid Application Development (RAD)* menjelaskan metode pengembangan perangkat lunak yang sangat menekankan pembuatan prototipe yang cepat dan pengiriman berulang. Oleh karena itu, model *RAD* merupakan alternatif yang tajam untuk model pengembangan air terjun yang khas, yang sering kali berfokus pada perencanaan dan praktik desain sekuensial. Pertama kali diperkenalkan pada tahun 1991 dalam buku James Martin dengan nama yang sama, pengembangan aplikasi yang cepat telah menjadi salah satu metode pengembangan yang paling populer dan kuat, yang termasuk dalam kategori teknik pengembangan tangkas orang tua.

2.2.1 Definisi

Rapid Application Development (RAD) adalah bentuk metodologi pengembangan perangkat lunak tangkas yang memprioritaskan rilis dan iterasi prototipe cepat. Berbeda dengan metode Waterfall, *RAD* menekankan penggunaan perangkat lunak dan umpan balik pengguna atas perencanaan yang ketat dan pencatatan persyaratan (Singh, 2019).

2.2.2 Manfaat

Ada beberapa manfaat menggunakan metode *RAD*. Beberapa manfaat dan keunggulan utama *RAD* antara lain:

- 1 Fleksibilitas dan kemampuan beradaptasi yang ditingkatkan karena pengembang dapat membuat penyesuaian dengan cepat selama proses pengembangan.
- 2 Iterasi cepat yang mengurangi waktu pengembangan dan mempercepat pengiriman.
- 3 Mendorong penggunaan ulang kode, yang berarti lebih sedikit pengkodean manual, lebih sedikit ruang untuk kesalahan, dan waktu pengujian yang lebih singkat.
- 4 Meningkatnya kepuasan pelanggan karena kolaborasi dan koordinasi tingkat tinggi antara pemangku kepentingan (pengembang, klien, dan pengguna akhir).
- 5 Manajemen risiko yang lebih baik karena pemangku kepentingan dapat mendiskusikan dan menangani kerentanan kode sambil menjaga proses pengembangan tetap berjalan.
- 6 Lebih sedikit kejutan karena, tidak seperti metode Waterfall, *RAD* menyertakan integrasi sejak awal dalam proses pengembangan perangkat lunak.

2.2.3 Tahapan

Ada beberapa tahapan yang dilakukan dalam melaksanakan metode *RAD*, antara lain:

1. Tentukan dan selesaikan persyaratan proyek

Selama langkah ini, para pemangku kepentingan duduk bersama untuk menentukan dan menyelesaikan persyaratan proyek seperti tujuan, harapan, jadwal, dan anggaran proyek. Jika Anda telah dengan jelas menentukan dan menentukan cakupan setiap aspek dari persyaratan proyek, Anda dapat meminta persetujuan manajemen.

2. Membuat prototipe

Segera setelah penyelesaian pelingkupan proyek, tahap berikutnya dapat memulai pengembangan. Desainer dan pengembang akan bekerja sama dengan klien untuk membuat dan meningkatkan prototipe yang berfungsi hingga produk akhir siap.

3. Kumpulkan masukan pengguna

Pada langkah ini, prototipe dan sistem beta diubah menjadi model kerja. Pengembang kemudian mengumpulkan umpan balik dari pengguna untuk mengubah dan meningkatkan prototipe dan membuat produk terbaik.

4. Pengujian produk

Bagian ini berfungsi untuk menguji produk perangkat lunak dan memastikan bahwa semua bagian yang bergerak bekerja sama sesuai harapan klien. Terus gabungkan umpan balik klien saat kode diuji dan diuji ulang untuk kelancaran fungsinya.

5. Implementasi sistem

Ini adalah langkah terakhir sebelum produk jadi diluncurkan. Ini melibatkan konversi data dan pelatihan pengguna.

2.2.4 Prototipe Pembuatan Prototipe

Sementara berbagai bentuk *RAD* menekankan konsep dan metodologi desain yang sedikit berbeda, penyertaan umum di sebagian besar sistem *RAD* adalah penggunaan prototipe yang berat. Sebagai alternatif dari spesifikasi desain yang rumit, penggunaan prototipe selama siklus pengembangan memberikan sejumlah manfaat unik:

2.2.5 Keterlibatan Pengguna

Tidak seperti model air terjun tradisional, yang mengharuskan tim desain untuk berdiskusi dengan pengguna fitur atau implementasi apa yang mungkin diperlukan dan spesifikasi rencana seputar ide-ide tersebut, prototipe cepat memungkinkan pengguna untuk benar-benar menggunakan perangkat lunak dan memberikan umpan balik pada sistem langsung, daripada mencoba memberikan evaluasi abstrak dari dokumen desain.

2.2.6 Kelayakan

Pembuatan prototipe memungkinkan tim pengembangan dengan cepat mengevaluasi kelayakan komponen yang sangat kompleks atau berisiko langsung dari awal. Dengan mengenali dan mengerjakan sistem yang rumit di awal siklus pengembangan, perangkat lunak akan lebih kuat, tidak terlalu rentan terhadap kesalahan, dan terstruktur lebih baik untuk penambahan desain di masa mendatang.

2.2.7 Pengurangan Kesalahan & Debugging

Dengan rilis prototipe cepat selama proyek, jauh lebih mungkin bahwa kesalahan akan ditemukan (dan kemudian dihilangkan) jauh lebih awal dalam siklus pengembangan daripada dengan pendekatan air terjun biasa.

2.2.8 Kelebihan Pengembangan Aplikasi yang Cepat

Meskipun ada sejumlah manfaat menggunakan metode pengembangan aplikasi cepat, kami telah menyoroti beberapa hal terpenting saat mempertimbangkan apakah *RAD* adalah pilihan yang tepat untuk proyek Anda berikutnya.

2.2.9 Kemajuan yang Dapat Diukur

Dengan seringnya iterasi, komponen, dan prototipe selesai, kemajuan pada keseluruhan proyek, serta segmen yang lebih kecil, dapat dengan mudah diukur dan dievaluasi untuk mempertahankan jadwal dan anggaran.

2.2.10 Menghasilkan Kode Produktif dengan Cepat

Karena sebagian besar pengembang perangkat lunak aktif beralih ke peran multidisiplin (yaitu pengembang tumpukan penuh), metodologi *RAD* memungkinkan anggota tim yang terampil untuk dengan cepat menghasilkan prototipe dan kode kerja untuk mengilustrasikan contoh yang mungkin membutuhkan waktu berminggu-minggu atau bulan untuk melihat cahaya hari menggunakan teknik pengembangan yang lebih lambat.

2.2.11 Kompartementalisasi Komponen Sistem

Mirip dengan praktik pemrograman berorientasi objek yang menjaga objek dan komponen dikarantina satu sama lain, *RAD* secara inheren memiliki dampak menguntungkan yang sama pada komponen yang dihasilkan selama pengembangan. Dengan memaksa perancang dan pengembang untuk membuat komponen yang berfungsi dan independen sendiri, untuk digunakan dalam rilis berulang atau prototipe, setiap elemen dalam sistem perangkat lunak secara keseluruhan dikelompokkan dan oleh karena itu mudah dimodifikasi seiring dengan perkembangan kebutuhan perangkat lunak.

2.2.12 Umpan Balik Pengguna yang Cepat dan Konstan:

Seperti dibahas di atas, mendapatkan umpan balik pengguna yang relevan selama pengembangan sangat berharga. Metodologi *RAD* memungkinkan antarmuka pengguna yang hampir konstan dan umpan balik melalui iterasi dan rilis prototipe yang sering, memberikan evaluasi dan kritik yang tak ternilai bagi seluruh tim saat paling dibutuhkan.

2.2.13 Integrasi Sistem Awal

Meskipun sebagian besar proyek perangkat lunak metode waterfall harus, pada dasarnya, menunggu hingga akhir siklus hidup untuk memulai integrasi dengan sistem atau layanan lain, aplikasi yang dikembangkan dengan cepat menjadi terintegrasi hampir seketika. Dengan mengharuskan integrasi awal dalam prototipe, sistem *RAD* dengan cepat mengidentifikasi kesalahan atau komplikasi apa pun dalam integrasi dan memaksa resolusi segera.

2.2.14 Adaptabilitas Sederhana

Selama pengembangan, perangkat lunak adalah bentuk yang cukup mudah dibentuk. Karena kode dapat diubah yang secara dramatis mengubah seluruh sistem atau menghasilkan komponen baru, tim pengembangan dapat memanfaatkan fleksibilitas ini sejak dini dan sering, dengan mengulang dan membuat prototipe konsep atau ide potensial selama pengembangan.

2.2.15 Kontra Pengembangan Aplikasi Cepat

Tidak ada metode pengembangan perangkat lunak yang tanpa kesalahan dan *RAD* juga demikian. Meskipun manfaatnya biasanya lebih besar daripada kerugiannya, kami akan memeriksa beberapa penghalang pandang yang paling mungkin terjadi saat menerapkan *RAD* ke dalam proyek baru.

2.2.16 Memerlukan Sistem Modular

Karena setiap komponen di dalam sistem harus dapat diulang dan dapat diuji dengan sendirinya, keseluruhan desain sistem saat menggunakan *RAD* mengharuskan setiap komponen menjadi modular, memungkinkan elemen untuk ditukar masuk dan keluar atau diubah oleh berbagai anggota tim.

2.2.17 Kesulitan Dalam Proyek Skala Besar

Meskipun metode pengembangan aplikasi yang cepat menghasilkan fleksibilitas yang jauh lebih besar selama proses desain dan pengembangan, metode ini juga cenderung mengurangi kontrol dan pembatasan. Meskipun ini

tidak secara inheren negatif, mengelola dengan benar fleksibilitas dan volatilitas tambahan ini dalam cakupan keseluruhan proyek bisa jadi sulit untuk aplikasi yang lebih besar.

2.2.18 Menuntut Antarmuka Pengguna yang Sering

Mendapatkan wawasan dan umpan balik pengguna lebih awal dan sering tentu saja merupakan manfaat dari perspektif desain, tetapi pedang bermata dua ini mengharuskan tim untuk bersedia dan mampu berkomunikasi dengan pengguna secara lebih bebas.

2.3 Apotek Setia Sehat Marelان

Apotek Setia Sehat terletak di Provinsi Sumatera Utara di Kota Medan dengan alamat Jl. Marelان Raya No. 195, Medan. Keperluan terhadap apotek dapat dilakukan dengan berhubungan langsung dengan apoteker. Surat Izin Praktik Apoteker (SIPA): No.445/52.33/II/2016 dan no Izin Apotek

2.4 *Unified Modeling Language*

Unified Modeling Language (UML) adalah bahasa grafis yang mirip dengan *Visio* atau *Balsamiq*. Itu dibuat untuk dengan mudah memvisualisasikan, menentukan, membangun dan mendokumentasikan sistem perangkat lunak berorientasi objek. *UML* digunakan dalam pemrograman dan dapat dihubungkan ke berbagai bahasa pemrograman seperti *Java*, *C*, *Visual Basic*, atau bahkan ditautkan ke database berorientasi objek.

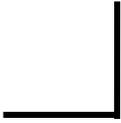
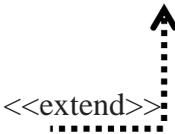
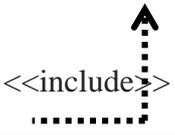
UML adalah standar pemodelan berorientasi objek yang dirancang oleh *Object Management Group*, sebuah grup yang didedikasikan untuk mengembangkan teknologi dan standar berorientasi objek sejak 1980-an. Sekarang, *UML* menjadi semakin populer di kalangan profesional *OOP* (*Object Oriented Programming*). *UML* juga digunakan untuk mengembangkan perkakas untuk desain berorientasi objek. *Unified Modeling Language (UML)*, dikembangkan oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson (Sukmawati & Priyadi, 2019).

2.4.1 *Use Case Diagram*

Diagram use case menunjukkan bagaimana berbagai pihak berinteraksi dengan sistem yang dikembangkan. Kasus penggunaan adalah pernyataan yang menjelaskan fungsionalitas sistem atau persyaratan yang harus dipenuhi sistem dari sudut pandang pengguna. Simbol-simbol dalam *Use Case Diagram* dapat dilihat pada tabel yang terlampir pada tabel 2.2.

Tabel 2.2 Elemen *Use Case Diagram*

SIMBOL	NAMA	KETERANGAN
	<i>Actor</i>	Menspesifikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i>
	<i>Use Case</i>	Deskripsi urutan aksi-sistem yang menghasilkan suatu hasil yang terukur
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas

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

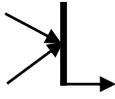
Sumber: (Kurniawan, 2018)

2.4.2 Activity Diagram

Activity Diagram atau diagram aktivitas adalah kombinasi dari pemodelan proses dan pemodelan cara. Diagram aktivitas memiliki beberapa elemen pemodelan sistem. Diagram aktivitas menunjukkan berbagai aktivitas dalam sistem yang sedang dirancang, bagaimana setiap aktivitas dimulai, kemungkinan keputusan yang terjadi, dan bagaimana penyelesaiannya. Diagram Aktivitas dapat mendeskripsikan beberapa peristiwa bersamaan yang mungkin terjadi dalam program juga. Diagram aktivitas adalah diagram status unik, dan transisi dipicu

dengan menyelesaikan status sebelumnya dan pemrosesan sendiri. Diagram Aktivitas tidak menjelaskan mekanisme internal dan hubungan internal antar komponen secara tepat. Diagram aktivitas menggambarkan berbagai alur kerja dan urutan kejadian, bagaimana aktivitas tertentu dimulai, keputusan yang mungkin terjadi dan bagaimana setiap aktivitas berakhir (Ladjamudin, 2017). Beberapa elemen *activity diagram* yang digunakan dapat dilihat pada tabel 2.3.

Tabel 2.3 Elemen Activity Diagram

SIMBOL	NAMA	KETERANGAN
	<i>Action State</i>	Menandakan sebuah aktivitas
	<i>Initial State</i>	Titik awal untuk memulai suatu aktivitas
	<i>Final State</i>	Titik akhir untuk mengakhiri aktivitas
	<i>Decision</i>	Pilihan untuk mengambil keputusan
	<i>Flow Final</i>	Untuk mengakhiri suatu aliran
	<i>Transition</i>	Menunjukkan aktifitas selanjutnya setelah aktivitas sebelumnya
	<i>Synchronization</i>	Dibagi menjadi 2 yaitu fork dan join: Fork digunakan untuk memecah behaviour menjadi activity atau action

		yang paralel, sedangkan join untuk menggabungkan kembali activity atau action yang paralel
	<i>Swimlane</i>	Untuk melakukan partisi atau pembagian
	<i>Signal Accept State</i>	Tanda penerimaan
	<i>Signal Send State</i>	Tanda pengiriman

Sumber: (Kurniawan, 2018)

2.4.3 *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar kelas dalam hal pertukaran pesan satu sama lain dari waktu ke waktu. Mereka disebut atrium. Diagram urutan adalah cara yang berguna untuk mensimulasikan banyak skenario waktu proses. Ini dapat membantu memprediksi bagaimana suatu sistem akan berperilaku dan digunakan dalam pemodelan sistem baru. Berikut komponen-komponen yang ada pada *Sequence Diagram*.

Tabel 2.4 Elemen *Sequence Diagram*

SIMBOL	NAMA	KETERANGAN
	<i>Objek</i>	Menggambarkan objek/orang yang berintraksi di dalam sistem
	<i>Stimulus</i>	Menggambarkan pengiriman pesan

	<i>Self Stimulus</i>	Menyatakan suatu objek mengirimkan pesan untuk menjalankan operasi yang ada pada objek lain.
---	----------------------	--

Sumber: (Kurniawan, 2018)

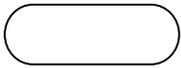
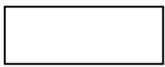
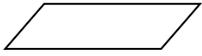
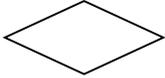
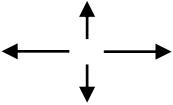
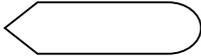
2.5 Diagram Alir

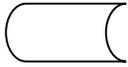
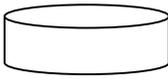
Diagram alir (*Flowchart*) adalah gambar langkah-langkah terpisah dari proses secara berurutan. Ini adalah alat umum yang dapat diadaptasi untuk berbagai tujuan dan dapat digunakan untuk menggambarkan berbagai proses, seperti proses pembuatan, proses administrasi atau layanan, atau rencana proyek. Ini adalah alat analisis proses yang relatif umum dan berguna dan salah satu dari tujuh alat kualitas yang diperlukan. Diagram alir adalah representasi visual dari urutan langkah dan keputusan yang diperlukan untuk melakukan suatu proses. Dalam urutan diagram adalah semua langkah yang telah dicatat. Langkah-langkahnya dihubungkan dengan garis dan panah, yang menunjukkan urutan berurutan. Alih-alih menuliskan beberapa proses yang panjang dan rumit, dengan *Hyperconcepts Flowchart*, seseorang dapat mengikutinya selangkah demi selangkah (Kurniawan, 2018).

Diagram alir adalah alat bisnis yang ampuh. Dapat menjelaskan bagaimana suatu proses dilakukan dengan jelas dan efisien dengan desain dan konstruksi yang tepat. Sebagaimana ditunjukkan dalam diagram alir, elemen yang disebutkan meliputi urutan tindakan, bahan atau layanan yang masuk atau keluar dari proses (masukan dan keluaran), keputusan yang harus dibuat, orang yang terlibat, waktu yang terlibat di setiap langkah, dan pengukuran proses.

Flowchart adalah diagram yang menggambarkan proses komputer, proses, atau sistem. Diagram digunakan secara luas di berbagai bidang seperti bisnis dan sains untuk merekam, mempelajari, merencanakan, meningkatkan, dan mengkomunikasikan konsep yang lebih kompleks dengan cara yang mudah dipahami. Diagram alir adalah representasi grafis dari suatu sistem atau langkah-langkah proses. Setiap desain menunjukkan struktur sistem, langkah-langkah yang diperlukan untuk menjalankan sistem, dan kondisi yang harus dipenuhi pada setiap langkah. Orang sering menggunakan ini untuk menggambarkan sistem atau proses. Bagan yang berbeda dapat berkisar dari bagan sederhana yang digambar dengan tangan yang menunjukkan langkah-langkah dan rute hingga diagram komprehensif yang digambar komputer yang menggambarkan beberapa langkah dan rute. Jika kita mempertimbangkan semua bentuk diagram alir, diagram tersebut mungkin salah satu diagram paling umum di planet ini, yang digunakan oleh orang-orang teknis dan non-teknis di berbagai bidang. Diagram Alir (atau Diagram Proses atau Peta Proses atau Model Proses Bisnis atau Diagram Alir Proses) terkadang disebut dengan nama yang lebih khusus, seperti Diagram Alir Proses, Peta Proses, diagram alir Fungsional, Pemetaan Proses Bisnis, atau Pemodelan Proses Bisnis. Ini mirip dengan diagram analisis lintas fungsi terkenal lainnya, seperti diagram aliran data dan diagram aktivitas bahasa pemodelan terpadu. Adapun simbol-simbol *Flowchart* lihat pada tabel sebagai berikut:

Tabel 2.5 Simbol *Flowchart*

NO	SIMBOL	FUNGSI
1.		Terminal , untuk memulai atau mengakhiri suatu program
2.		Proses , suatu simbol yang menunjukkan setiap pengolahan yang dilakukan.
3.		Input-Output , untuk memasukkan menunjukkan hasil dari suatu proses
4.		Decision , suatu kondisi yang akan menghasilkan beberapa kemungkinan jawaban atau pilihan
5.		Preparation , suatu symbol yang menyediakan tempat pengolahan
6.		Connector , suatu prosedur penghubung yang akan masuk atau keluar melalui symbol ini dalam lembar yang sama
7.		Off-Page Connector , merupakan symbol masuk atau keluarannya suatu prosedur pada lembaran kertas lainnya
8.		Arus/Flow , dari pada prosedur yang dapat dilakukan atas ke bawah dari bawah ke atas, ke atas dari kiri ke kanan ataupun dari kanan ke kiri
9.		Predefined Process , untuk menyatakan sekumpulan langkah proses yang ditulis sebagai prosedur
10.		Simbol untuk <i>output</i> , yang ditunjukkan ke suatu <i>device</i> , seperti printer, dan sebagainya

11		Penyimpanan <i>file</i> secara sementara
12		Menunjukkan <i>input / Output Hardisk</i> (media penyimpanan)

Sumber: (Kurniawan, 2018)

2.6 Pemrograman

2.6.1 Bahasa Pemrograman

Bahasa pemrograman adalah suatu bentuk bahasa yang terdiri dari sekumpulan instruksi yang menghasilkan jenis keluaran tertentu. Bahasa pemrograman teknologi tinggi digunakan dalam pemrograman komputer untuk menerapkan algoritma teknologi tinggi. Kebanyakan bahasa pemrograman melibatkan instruksi yang dapat dimengerti oleh komputer. Selain itu, mesin yang dapat diprogram menggunakan sekumpulan instruksi tertentu; daripada menggunakan bahasa pemrograman umum. Program telah digunakan untuk mengarahkan perilaku mesin, seperti piano pemain. Meskipun cara tangan bergerak di atas keyboard bervariasi tergantung pada apa yang dikirimkan oleh program, itu tidak menghasilkan perilaku yang berbeda dalam menanggapi input atau kondisi yang berbeda (Gabbrielli & Martini, 2010).

Ribuan bahasa pemrograman yang berbeda telah dibuat untuk pemrograman, dan yang baru dibuat setiap tahun. Banyak bahasa pemrograman ditulis dalam gaya imperatif (yaitu, sebagai urutan operasi yang harus dilakukan). Sebaliknya, yang lain adalah gaya yang lebih deklaratif (yaitu, sebagai rangkaian

deklarasi) (yaitu, hasil yang diinginkan ditentukan, bukan bagaimana mencapainya).

Deskripsi bahasa pemrograman biasanya memerlukan dua komponen: sintaks, yang mendeskripsikan format untuk data dan maksud, dan semantik, yang mendeskripsikan arti data dan bagaimana data itu diproses (makna). Beberapa bahasa ditentukan oleh dokumen spesifikasi (misalnya, bahasa pemrograman C ditentukan oleh Standar *ISO*). Sebaliknya, bahasa lain (seperti *Perl*) memiliki implementasi dominan yang diperlakukan sebagai referensi. Bahasa asli seperti bahasa Inggris memiliki keduanya. Sebuah standar mendefinisikan bahasa utama, dan ekstensi ditambahkan ke implementasi dominan.

2.6.2 Pemrograman *Web*

Pemrograman adalah seni rumit untuk memberi tahu komputer apa yang harus dilakukan. Secara logis memberikan instruksi tertulis yang dapat dimengerti komputer. Pada dasarnya, Anda memberikan instruksi kecil kepada komputer, dan komputer turun daftar, melakukan masing-masing secara berurutan. Pemrograman memungkinkan Anda membuat perangkat lunak baru dan membuat komputer melakukan hal-hal baru. Pemrograman situs *web* sama kecuali Anda menulis aplikasi atau halaman *web* yang digunakan oleh *browser web*.

Keuntungannya adalah bahwa pemrograman situs *web* bisa jadi mudah. Pemrograman *web* adalah praktik menulis aplikasi yang berjalan di *server web* dan dapat digunakan oleh banyak orang yang berbeda. Banyak aplikasi bekerja dengan baik sebagai aplikasi *web*. Beberapa contoh aplikasi web populer termasuk

Flickr, *Gmail*, dan *Google Maps*. Seseorang dapat mengunggah dan melihat gambar, mengirim email dan mencari petunjuk menggunakan aplikasi *web* ini, dan semuanya melalui pemrograman *web* (Nugroho, 2018).

2.6.3 Desain Web

Sebuah desain *web* terdiri dari grafik, animasi dan dapat mempercantik *website*. Desain web adalah bentuk pembuatan halaman *web*. Banyak yang belum mengetahui arti desain web. Sebagian orang menganggap desain *website* hanyalah media *online*. Dengan perkembangan teknologi dan kemajuan internet, situs berita bermunculan di seluruh dunia. Di sinilah desain interaktif memainkan peran kunci. Penting untuk mengikuti desain *web* praktik yang baik. Desain web harus mudah digunakan. Pengguna benar-benar tidak menginginkan desain yang rumit dan cenderung tersesat dalam menemukan berbagai informasi tentang produk atau layanan Anda. Membuat situs *web* seperti membuat rumah sendiri.

2.6.4 PHP

PHP adalah bahasa pemrograman opsional, sangat cocok untuk halaman web. Skrip *PHP* akan dijalankan di server web oleh juru bahasa *PHP* yang dimuat sebagai skrip modul, daemon, atau *CGI*. Ini akan menjadi bagian dari respons *HTTP* di *server web*. Ada banyak sistem template, sistem manajemen konten, dan kerangka kerja yang dapat mengaktifkan respons. *PHP* juga dapat digunakan untuk membuat program lain seperti program grafis dan program untuk robotika

seperti drone. Anda juga dapat menjalankan skrip *PHP* langsung dari baris perintah.

Perangkat lunak *PHP* gratis tersedia di bawah *Lisensi Atribusi Creative Commons*. *PHP* telah tersedia secara luas dan terbuka untuk digunakan. *PHP* adalah bahasa pemrograman *open-source* yang berkembang dengan cepat yang dikembangkan tanpa spesifikasi atau standar resmi. Pekerjaan telah ditetapkan untuk protokol *PHP* formal. Pada Januari, lebih dari 72 persen situs *web PHP* menggunakan versi *PHP* yang dihentikan, yaitu *PHP 7.2* tidak lagi didukung oleh Tim Pengembang *PHP*. Telah diketahui bahwa sekitar 40% dari semua situs *web* menggunakan *PHP 7.3*, yang hanya (hingga 6 Desember 2021) "hanya didukung untuk masalah keamanan kritis". Ada sekitar 42% situs *web* yang menggunakan *PHP 5.6* atau lebih lama (Kustiyaningsih & Devie, 2017).

2.6.5 Cascading Style Sheets

Cascading Style Sheets, disingkat *CSS*, adalah bahasa desain sederhana yang dimaksudkan untuk menyederhanakan proses penerbitan situs web. *CSS* mendefinisikan tampilan dan struktur halaman *web*. Dengan menggunakan *CSS*, Anda dapat memanipulasi tampilan visual teks halaman *web* Anda dengan mudah. *CSS* mudah dipahami dan dipelajari, tetapi *CSS* menyediakan banyak opsi rendering penting. Paling umum, *CSS* digunakan dengan *HTML* atau *XHTML*. Kode *CSS* dapat ditulis sekali dan digunakan kembali pada lembar yang sama di beberapa halaman *HTML*. Anda dapat menerapkan definisi gaya untuk setiap elemen *HTML* dan menerapkannya ke halaman sebanyak yang Anda inginkan.

Terapkan aturan *CSS* yang sama untuk setiap kemunculan tag. Dengan demikian kurangnya kode berarti waktu unduh lebih cepat (Saputra & Agustin, 2011).

Untuk membuat perubahan besar pada seluruh situs web, ganti templatnya, dan semua halaman akan diperbarui secara otomatis. Dibandingkan dengan *CSS*, *HTML* memiliki variasi fitur yang lebih luas; programmer dapat memberikan tampilan yang lebih baik ke dokumen *HTML* Anda. Kompatibilitas beberapa perangkat menyiratkan kemampuan untuk menyesuaikan gaya untuk perangkat yang berbeda. Dengan dokumen *HTML* yang sama, versi situs web serupa cocok untuk perangkat genggam seperti *PDA* dan ponsel atau pencetakan. Sekarang disarankan untuk menggunakan *CSS* daripada atribut *HTML*. Oleh karena itu, sebaiknya tingkatkan kompatibilitas situs Anda menggunakan *CSS* di halaman *HTML*.

CSS dibuat dan dikelola oleh kelompok komunitas *W3C* yang disebut Kelompok Kerja *CSS*. Kelompok Kerja *CSS* membuat dokumen yang disebut dokumen yang disebut spesifikasi. Ketika spesifikasi dilengkapi dan ditandatangani oleh anggota *W3C*, itu menjadi rekomendasi resmi. Ini disebut rekomendasi karena *W3C* tidak memiliki kendali atas implementasi bahasa atau standar yang sebenarnya. Perusahaan independen dan organisasi nirlaba mengembangkan perangkat lunak itu. *World Wide Web Consortium (W3C)* adalah grup yang menentukan laju perkembangan Internet.

CSS level 1 direkomendasikan oleh *W3C* pada bulan Desember 1996. *CSS* dan *HTML* dibahas secara rinci dalam versi ini. Pada Mei 1998, *CSS2* menjadi

rekomendasi W3C. *World Wide Web Accessibility (WCAG) Versi 2.0* kompatibel dengan Pedoman Aplikasi Internet Kaya yang Dapat Diakses (*WAI-ARIA*).

2.7 Database

Database adalah kumpulan informasi terstruktur (data) yang terorganisir, disimpan dalam sistem komputer. Database dikendalikan oleh sistem manajemen database, biasanya (*DBMS*). Sistem basis data adalah kombinasi dari sistem manajemen data dan sistem manajemen basis data. Baris dan kolom mengatur data di sebagian besar *database* modern untuk kemudahan penggunaan dan pengambilan. Data yang dikumpulkan dapat diakses, dimodifikasi, diperbarui, dikontrol, dan diatur. Kebanyakan database menggunakan *SQL* untuk menulis dan membuat kueri data.

2.7.1 Structured Query Language

Structured Query Language (SQL) adalah bahasa pemrograman yang digunakan untuk menanyakan, memanipulasi, dan mendefinisikan data dalam database relasional. Meskipun *SQL* pertama kali dikembangkan di *IBM* pada 1970-an, *SQL* diikuti oleh organisasi seperti *Oracle* dan *Microsoft* pada tahun-tahun berikutnya. Meskipun *SQL* masih menjadi bahasa pemrograman standar saat ini, teknologi lain mulai menyusul.

SQL dapat digunakan untuk berbagi dan mengelola data, khususnya data yang ditemukan dalam sistem manajemen basis data relasional, termasuk data yang disusun ke dalam tabel. Beberapa file, masing-masing berisi tabel data, juga

dapat dihubungkan bersama oleh bidang yang sama. Dengan menggunakan *SQL*, Anda dapat membuat kueri, memperbarui, dan mengatur ulang data serta membuat dan mengubah skema (struktur) sistem *database*, dan mengontrol akses ke datanya (Karlos et al., 2016).

2.7.2 Evolusi Database

Database telah berkembang pesat sejak didirikan pada tahun 1960-an. *Database* navigasi seperti database hierarki dan jaringan adalah sistem pemodelan pertama yang digunakan untuk menyimpan dan memanipulasi data. Namun, sistem awal ini lebih rentan. *Database* relasional menjadi populer pada 1980-an, diikuti oleh database berorientasi objek pada 1990-an. Sejak munculnya internet, *NoSQL* dan sistem basis data lainnya menjadi lebih terkenal. *Database cloud* dan *database self-driving* bekerja untuk merevolusi data.

2.8 Tipe Data

Komputer hanya diprogram untuk menerima angka sebagai input. Mereka tidak mengenali singkatan. Ada tipe data fundamental di sebagian besar bahasa pemrograman. Setelah aplikasi komputer dirancang, pemrogram harus memastikan bahwa tipe data yang mereka gunakan direferensikan dengan benar. Beberapa tipe data juga digunakan dalam bahasa pemrograman yang berbeda untuk alasan yang berbeda. Memahami berbagai jenis data memungkinkan pemrogram merancang program komputer dengan lebih efisien dan akurat.

Beberapa contoh umum tipe data, antara lain:

1. *Boolean* (contohnya: True or False)
2. *Character* (contohnya: a)
3. *Date* (contohnya: 03/01/2016)
4. *Double* (contohnya: 1.79769313486232E308)
5. *Floating-point number* (contohnya: 1.234)
6. *Integer* (contohnya: 1234)
7. *Long* (contohnya: 123456789)
8. *Short* (contohnya: 0)
9. *String* (contohnya: abcd)
10. *Void* (contohnya: no data)

2.8.1 Tipe Data Numeric

Data numerik adalah tipe data yang paling penting. Selain itu, data numerik selalu dikumpulkan dalam bentuk angka. Data numerik adalah metode paling berharga untuk mengidentifikasi angka. Tipe data *numeric* dibedakan dalam dua macam, yaitu *integer* dan *floating point*. *Integer* digunakan untuk data bilangan bulat sedangkan *floating point* digunakan untuk bilangan desimal. Tipe data numerik selengkapnya dapat dilihat pada tabel 2.6.

Tabel 2.6 Tipe Data Numeric

Tipe Data	Kisaran Nilai
<i>TINYINT</i>	(-128)-127 atau (0-255)
<i>SMALLINT</i>	(-32768)-32767 atau (0-65535)
<i>MEDIUMINT</i>	(-3888608)-8388607 atau 0-16777215
<i>INT, INTEGER</i>	(-2147683648)-(-21447683647) atau 0-4294967295
<i>FLOAT</i>	(-3.4 E+38)-(-1.17E-38), 0 dan 1.175E-38-3.4e+38
<i>DOUBLE</i>	(-1.79E+308)-(-2.225E-308), 0 dan 2.225E-308 – 1.79E+308

Sumber: (Ramadhani, 2019)

2.8.2 Tipe Data String

String dikenal sebagai suatu kalimat atau kata-kata yang berhubungan dengan penjelasan suatu benda. *String* adalah kumpulan beberapa karakter. Tipe-tipe data yang termasuk dalam tipe data *string* dapat dilihat pada tabel 2.7.

Tabel 2.7 Tipe Data String

Tipe Data	Kisaran Nilai
<i>CHAR</i>	1-255 karakter
<i>VARCHAR</i>	1-255 karakter
<i>TINYTEXT</i>	1-255 karakter
<i>TEXT</i>	1-65535 karakter
<i>MEDIUMTEXT</i>	1-16777215 karakter
<i>LONGTEXT</i>	1-424967295 karakter

Sumber: (Ramadhani, 2019)

2.8.3 Tipe Data *Date*

Tipe data *data* adalah tipe data untuk merepresentasikan tanggal dan jam. Contohnya adalah *DATETIME*, *DATE*, *TIMESTAMP*, *TIME* dan *YEAR*. Tiap tipe data memiliki nilai tertentu. Bahasa pemrograman akan memberikan peringatan kesalahan apabila tanggal atau waktu yang dimasukkan tidak sesuai format. Besaran nilai dan besar memori penyimpanan yang diperlukan untuk masing-masing tipe dapat dilihat pada tabel 2.8.

Tabel 2.8 Tipe Data *Date*

Tipe Data	Kisaran Nilai
<i>DATETIME</i>	1000-01-01 00:00 sampai 9999-12-31 23:59:59
<i>DATE</i>	1000-01-01 sampai 9999-12-31
<i>TIMESTAMP</i>	1970-01-01 00:00:00 sampai 2037
<i>TIME</i>	-839:59:59 sampai 838:59:59
<i>YEAR</i>	1901 sampai 2155

Sumber: (Ramadhani, 2019)

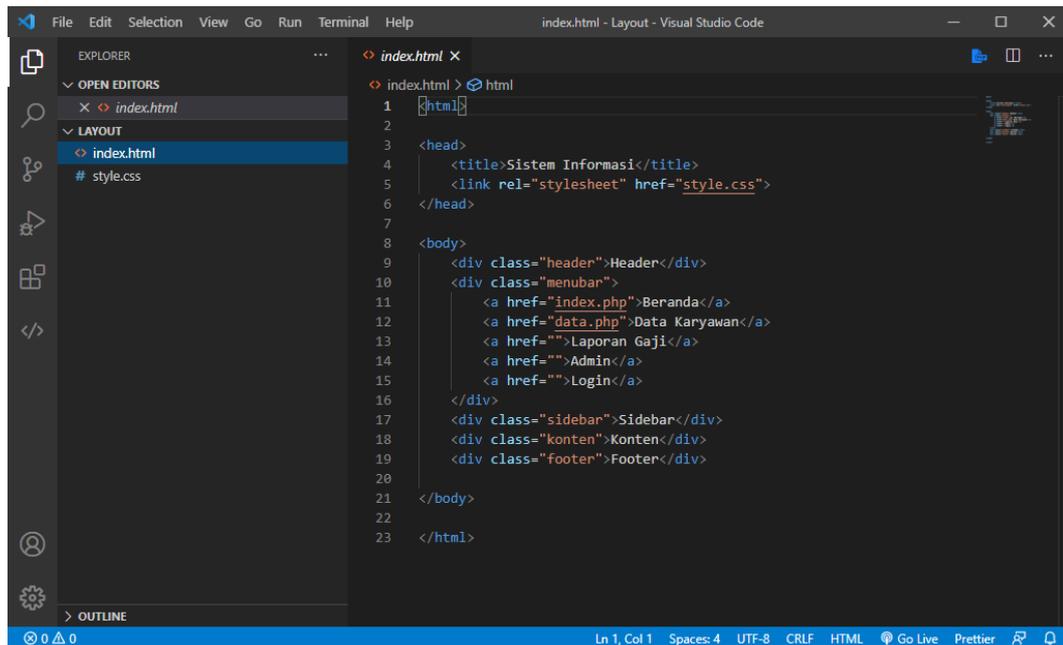
2.9 *Visual Studio Code*

Visual Studio Code adalah editor kode sumber yang ringan namun kuat yang berjalan di desktop Anda dan tersedia untuk *Windows*, *macOS*, dan *Linux*. Muncul dengan dukungan *built-in* untuk *JavaScript*, *TypeScript*, dan *Node.js* dan memiliki ekosistem ekstensi yang kaya untuk bahasa lain (seperti *C ++*, *C #*, *Java*, *Python*, *PHP*, dan *Go*) dan *runtime* (seperti *.Net* dan *Kesatuan*).

Selain dari keseluruhan gagasan untuk menjadi ringan dan memulai dengan cepat, *VS Code* memiliki penyelesaian kode *IntelliSense* untuk variabel, metode, dan modul yang diimpor; *debugging* grafis; linting, pengeditan multi-kursor, petunjuk parameter, dan fitur pengeditan canggih lainnya; navigasi kode yang manis dan pemfaktoran ulang; dan kontrol kode sumber bawaan termasuk dukungan *Git*. Sebagian besar diadaptasi dari teknologi *Visual Studio*.

VS Code yang tepat dibangun menggunakan *Electron shell*, *Node.js*, *TypeScript*, dan protokol Server Bahasa dan diperbarui setiap bulan. Ekstensi diperbarui sesering yang diperlukan. Kekayaan dukungan bervariasi di berbagai bahasa pemrograman dan ekstensinya, mulai dari penyorotan sintaks sederhana dan pencocokan braket hingga *debugging* dan *refactoring*. Anda dapat menambahkan dukungan dasar untuk bahasa favorit Anda melalui pewarna *TextMate* jika tidak ada server bahasa yang tersedia (Heller, 2019).

Kode di repositori *Visual Studio Code* adalah *open source* di bawah Lisensi MIT. Produk *VS Code* itu sendiri dikirimkan di bawah lisensi produk *Microsoft* standar, karena memiliki persentase kecil dari kustomisasi khusus *Microsoft*. Ini gratis meskipun ada lisensi komersial. Gambar 2.3 adalah editor *Visual Studio Code* (Heller, 2019).



Gambar 2.1 Editor *Visual Studio Code*

2.10 *World Wide Web*

2.10.1 Definisi *Web*

World Wide Web, atau biasa disebut sebagai *Web*, adalah sistem yang saling berhubungan yang memungkinkan publik untuk mengakses informasi mereka melalui Internet dengan mudah. Tidaklah benar untuk percaya bahwa *Web* itu sama dengan Internet. *Web* dapat dilihat sebagai salah satu dari banyak aplikasi yang dibangun di atas Internet (Jovan, 2007).

Pada tahun 1989, Tim *Berners-Lee* mengusulkan arsitektur dari apa yang akan menjadi *World Wide Web*. Dia adalah orang pertama yang membuat server *web*, *browser web*, dan halaman *web* pertama melalui komputernya di *CERN* (Laboratorium Fisika Partikel Eropa) pada tahun 1990. Pada tahun 1991, seorang

pria bernama Tim *Berners-Lee* mengumumkan pembuatan dan pengenalan *World Wide Web* ke dunia luas.

Internet terdiri dari berbagai alat terkait *web* seperti *browser*, *email*, dan obrolan.

1. Protokol *HTTP* mengatur bagaimana data ditransfer antara server dan klien.
2. Untuk mengakses komponen *Web*, klien menyediakan pengenal universal unik, yang disebut *URL* (lokasi sumber daya seragam) atau *URL* (pengenal sumber daya seragam) atau disebut Pengenal Dokumen Universal.
3. *HTML* (*hypertext markup language*) adalah format paling umum untuk menerbitkan dokumen *web*.

Hyperlinking, atau *Linking*, adalah suatu konsep yang mendefinisikan *web* sebagai kumpulan dokumen yang terhubung, sehingga dapat dikatakan bahwa *Web* adalah kumpulan dokumen yang terhubung.

Setelah menemukan *WWW* (*World Wide Web*), Tim *Berners-Lee* kemudian membuat *W3C* (*World Wide Web Consortium*) untuk mempromosikan *Web* lebih jauh. Konsorsium ini terdiri dari banyak grup kepentingan *web* inti, termasuk pengembang *browser web*, entitas pemerintah, grup penelitian ilmiah, dan universitas. Organisasi memiliki misi ganda, pendidikan dan penjangkauan.

2.10.2 Server *Web*

Server *web* menyimpan dan menyampaikan informasi yang diminta klien. Jenis klien yang paling umum adalah *browser web* yang meminta informasi dari situs *web* Anda dengan mengklik tautan atau mengunduh dokumen pada suatu halaman di browser. *Hypertext Transfer Protocol* berkomunikasi dengan *browser web* (*HTTP*). Dalam bahasa *markup Hypertext*, sebagian besar halaman *web* dikodekan dalam konten (*HTML*). Isinya mungkin statis atau dinamis (misalnya, teks, dan gambar) (misalnya, harga yang dihitung atau daftar item yang telah ditandai pelanggan untuk dibeli). Sebagian besar *server web* mendukung skrip bahasa sisi *server* untuk menyandikan logika bisnis dalam komunikasi untuk mengirimkan konten dinamis. *Active Server Pages (ASP)*, *JavaScript*, *PHP*, *Python*, dan *Ruby* adalah bahasa yang paling sering didukung.

Server web juga dapat menyimpan konten dalam *cache* untuk mengirimkan konten yang biasanya diminta dengan cepat. Ini juga disebut percepatan *web* (Pratama, 2010).

Sumber daya perangkat lunak dan perangkat keras yang sama yang dikenal sebagai *hosting* virtual dapat menjadi tuan rumah satu situs *web* atau beberapa situs. *Server web* juga dapat membatasi kecepatan reaksi berbagai pelanggan sehingga satu klien tidak mendominasi sumber daya yang paling baik digunakan untuk memenuhi permintaan dari sejumlah besar pelanggan.

Meskipun *server web* biasanya meng-*host* situs *web* yang diakses Internet, mereka juga dapat digunakan di jaringan lokal seperti intranet perusahaan untuk berkomunikasi antara klien *web* dan *server*. Dalam kamera digital seperti itu,

server web dapat dimasukkan ke dalam perangkat untuk memungkinkan pengguna berkomunikasi dengan perangkat melalui *browser Web* standar apa pun.

2.10.3 Situs Web

Situs *web* adalah dokumen dalam *Hypertext Markup Language* yang dapat dilihat di browser web. Sebuah alamat *URL* dapat dimasukkan ke dalam bilah alamat *browser web*. Situs *web* dapat menyertakan teks dan gambar serta tautan ke situs *web* lain (Pratama, 2010).

Sebuah situs *web* memberikan informasi kepada pengguna akhir, termasuk gambar atau video yang terkait dengan topik untuk membantu mengilustrasikan. Sebuah situs *web* juga dapat digunakan untuk menjual barang atau jasa kepada klien potensial. *Web* terdiri dari beberapa halaman yang bersama-sama membentuk sebuah situs *web*.

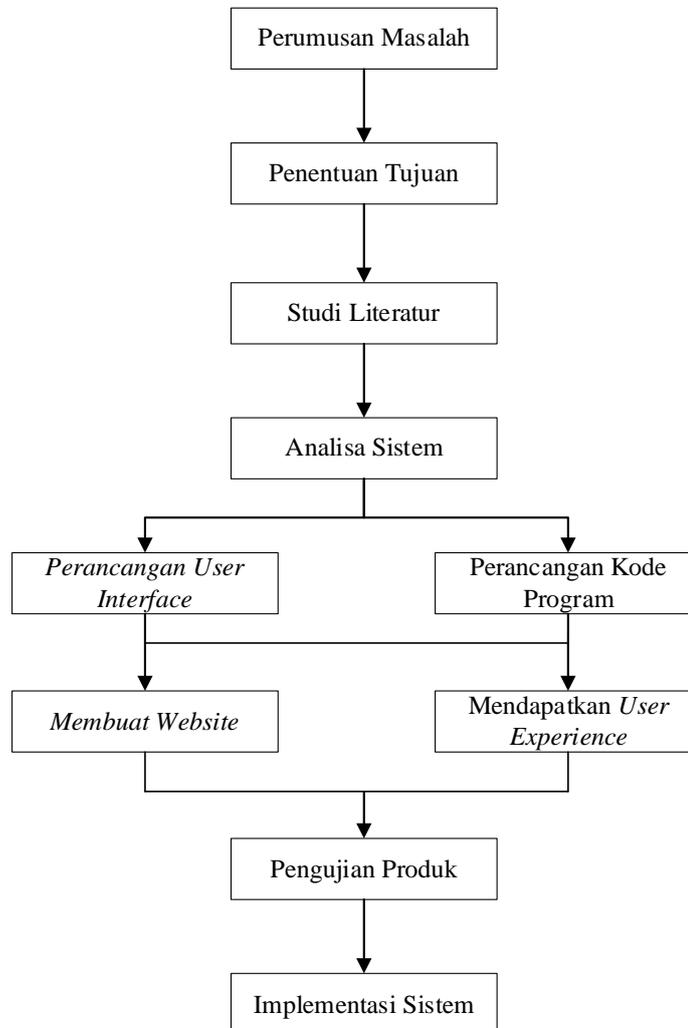
Saat Anda mengklik link di layar mesin pencari, Anda diarahkan ke sebuah situs *web*. Internet terdiri dari banyak halaman yang tersedia *online*, dengan lebih banyak lagi dibuat setiap hari.

BAB III

METODE PENELITIAN

3.1 Tahapan Penelitian

Penelitian ini menggunakan metode *Rapid Application Development (RAD)* yang dapat dilihat pada gambar 3.1



Gambar 3.1 Tahapan Penelitian

Gambar 3.1 menjelaskan bagan yang akan dicapai dalam merancang penelitian. Tahapan disusun menggunakan metode *Rapid Application Development (RAD)*. Hal ini bertujuan agar tidak terjadi penyimpangan dalam melaksanakan penelitian tersebut. Tahapan berikut adalah prosedur yang diambil dalam melakukan penelitian ini:

1. Perumusan masalah merupakan bagaimana merumuskan suatu masalah yang berasal dari identifikasi masalah sebelumnya.
2. Penentuan tujuan merupakan harapan dari hasil penelitian yang akan dikerjakan.
3. Studi literatur merupakan pembelajaran secara teoritis dalam mencari jawaban berdasarkan pencarian referensi yang berhubungan dengan metode *RAD*. Pembelajaran dapat berasal dari buku, jurnal, dan internet yang berhubungan dengan metode *RAD* tersebut.
4. Analisa sistem dilakukan untuk melihat sistem yang sebelumnya digunakan, mencari tahu kelemahan sistem tersebut dan mengembangkan sesuai dengan target yang akan dicapai. Perancangan sistem juga dilakukan berdasarkan informasi yang diperoleh dalam menentukan bagian dari sistem yang akan dibangun.
5. Perancangan *user interface* bertujuan untuk membuat desain sistem yang menampilkan antarmuka terhadap pengguna.
6. Perancangan kode program adalah bagian dalam membangun suatu sistem yang berupa *website* yang akan berhubungan langsung dengan pengguna.

7. Pembuatan *website* merupakan pembuatan uji coba sistem informasi pertama untuk mendapatkan hasil sementara dan mencari tahu kelemahan.
8. Mendapatkan *user experience* berfungsi mengumpulkan informasi dari pengguna sistem dan berusaha mengembangkan sistem tersebut berdasarkan testimoni pengguna.
9. Pengujian produk dilakukan dengan cara melakukan penetapan sistem secara online untuk melihat perkembangan sistem tersebut.
10. Implementasi sistem berfungsi untuk menerapkan sistem yang telah dihasilkan secara permanen. Implementasi dilakukan untuk mendapatkan manfaat sistem bagi pengguna dan pengembang sistem tersebut.

3.2 Metode Pengumpulan Data

Pengumpulan data diperoleh dari Apotek Setia Sehat. Data yang terjadi adalah proses penjualan akan digunakan untuk melakukan uji sistem informasi yang dibangun. Pengujian dilakukan agar dapat menghasilkan sistem yang benar dan sesuai dengan yang diharapkan.

Berikut ini adalah tahapan pengumpulan data yang ada pada penelitian dalam mendapatkan data, antara lain:

1. Studi Kepustakaan

Studi kepustakaan merupakan hal yang dilakukan untuk mencari sumber-sumber referensi yang ada sebagai bahan pertimbangan dalam membuat sistem informasi. Metode pengumpulan data ini dilakukan dengan cara mempelajari, membaca dan mencari beragam referensi yang ada baik itu

buku, jurnal, makalah, dan lain sebagainya sehingga dapat diambil dan dirangkup menjadi data uji yang sesuai dengan kebutuhan.

2. Wawancara

Wawancara dilakukan kepada apoteker terhadap obat-obat non generik yang dijual di apotek tersebut. Wawancara bertujuan untuk menentukan deskripsi dan spesifikasi obat non generik yang akan dimasukkan ke dalam sistem informasi.

3. Pengamatan

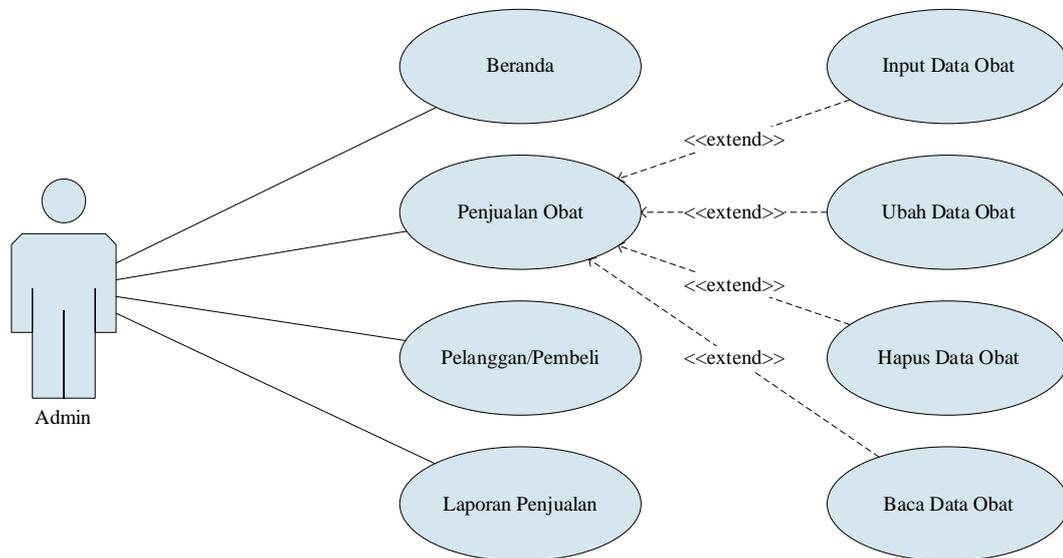
Pengamatan dilakukan secara langsung tentang kegiatan yang terjadi di Apotek Setia Sehat dalam melakukan penjualan obat. Pengamatan dilakukan pada waktu kegiatan penjualan obat berlangsung.

3.3 Pemodelan Penelitian

Penelitian memiliki beberapa model yang berbentuk diagram untuk menggambarkan secara langsung kegiatan dari sistem informasi yang akan dibangun dalam penelitian ini. Ada tiga buah diagram yang digunakan dalam penelitian ini, yaitu *Use Case Diagram*, *Activity Diagram* dan *Sequence Diagram*.

3.3.1 Use Case Diagram

Use Case Diagram menggambarkan kerja dari sistem informasi dari sudut pandang pengguna. Diagram ini memiliki cara kerja dengan cara menjelaskan hubungan interaksi antara pengguna dan sistem. Gambar 3.2 adalah perancangan *Use Case Diagram* pada penelitian ini.

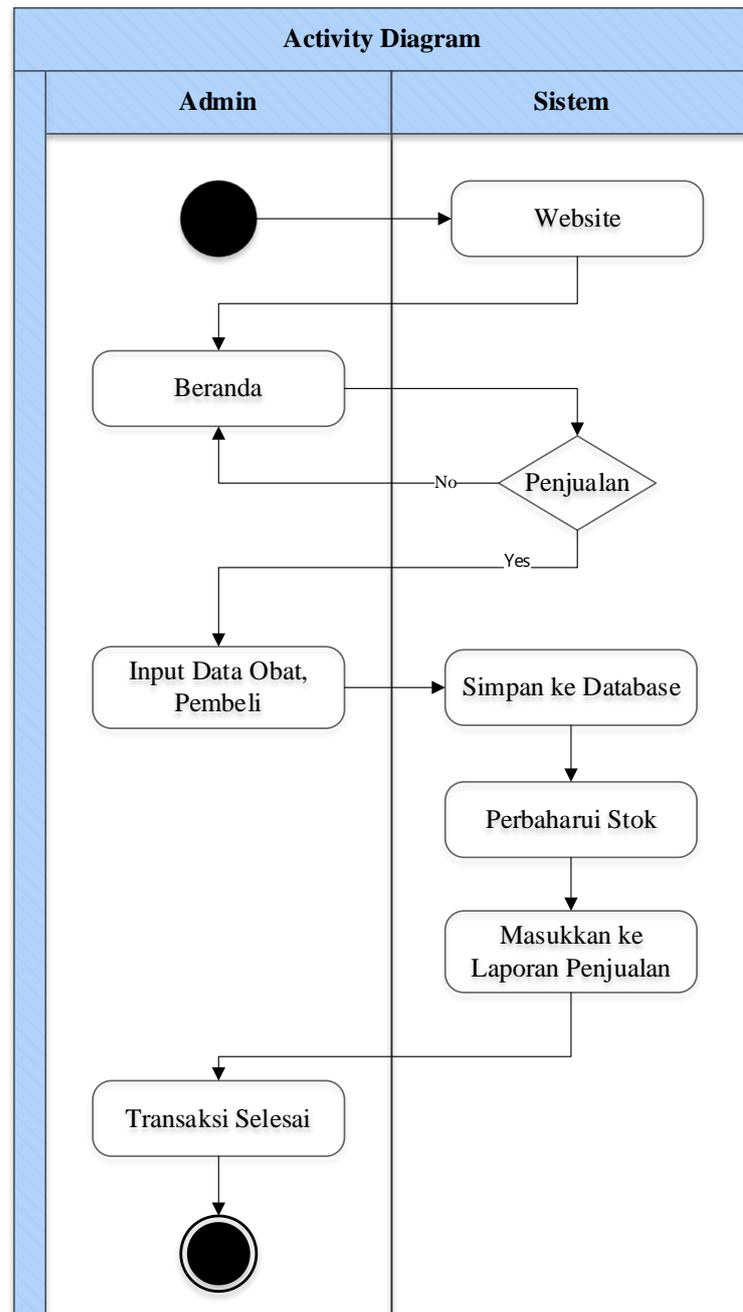


Gambar 3.2 Use case diagram penjualan obat

Gambar 3.2 menjelaskan bahwa *admin* yang bertindak sebagai pengguna dapat melakukan beberapa fungsi. Fungsi utama yang digunakan dalam sistem adalah penjualan obat. Pada proses penjualan, *admin* dapat memasukkan data produk, pelanggan dan laporan penjualan. Pihak apotek akan mendapatkan informasi lengkap terhadap penjualan obat non generik tersebut.

3.3.2 Activity Diagram

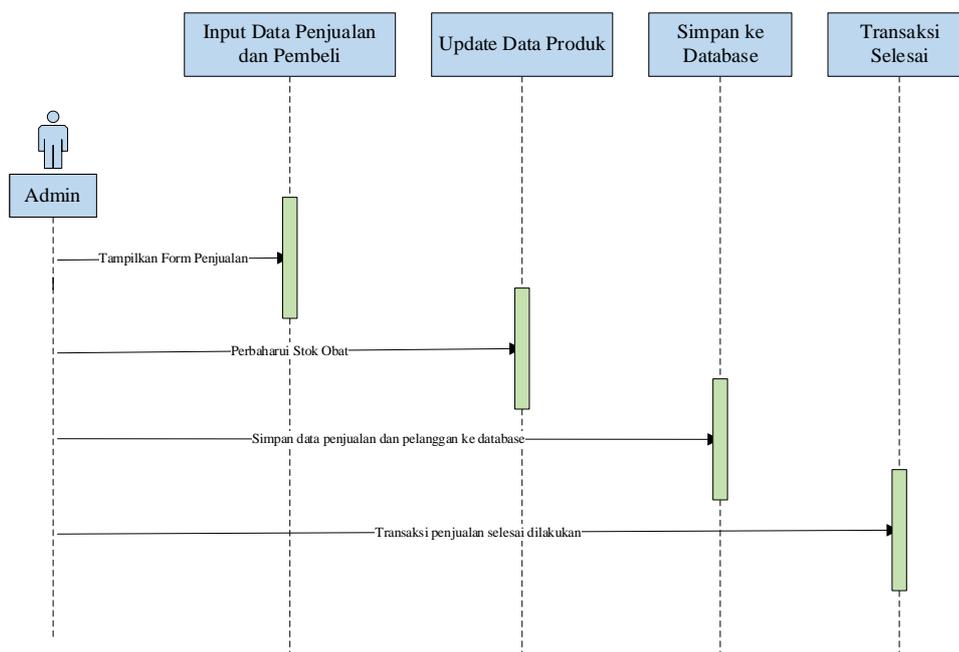
Activity Diagram menjelaskan kegiatan yang dilakukan oleh pengguna terhadap sistem tersebut. Diagram ini akan menjelaskan bagaimana proses sistem informasi berinteraksi dengan pengguna. Gambar 3.3 merupakan *Activity Diagram* dari sistem yang dirancang.



Gambar 3.3 Activity diagram penjualan obat

3.3.3 Sequence Diagram

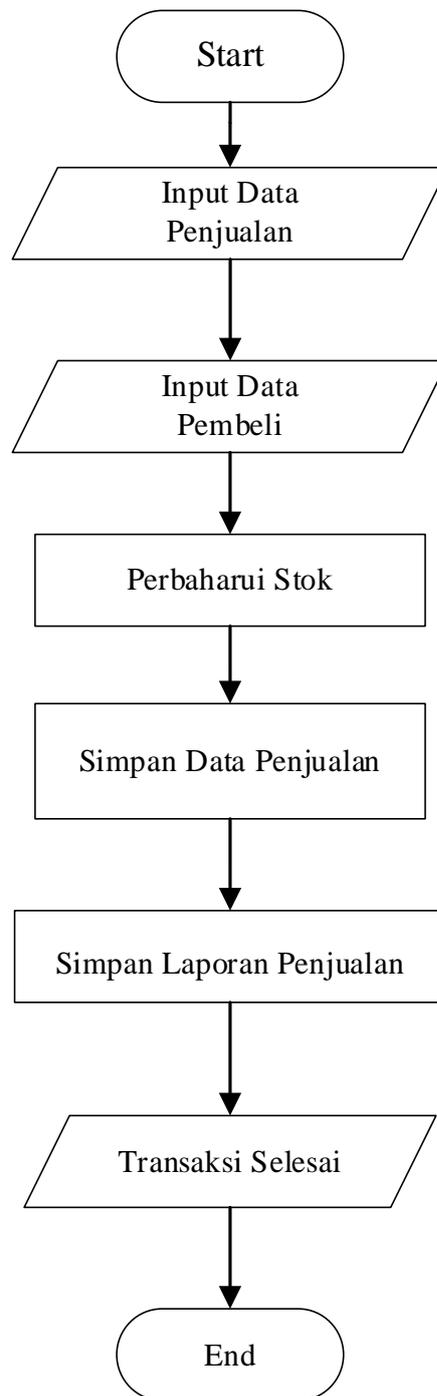
Sequence Diagram merupakan urutan kegiatan yang dilakukan *user* dalam menjalankan sistem informasi penjualan obat. Gambar 3.4 adalah *sequence diagram* yang digunakan pada penelitian ini.



Gambar 3.4 *Sequence diagram* penjualan

3.3.4 Flowchart

Flowchart menjelaskan proses dari penjualan obat di Apotek Setia Sehat. Proses penjualan meliputi input data penjualan, data pembeli dan pembaharuan stok obat yang ada di apotek. Gambar 3.5 adalah *flowchart* sistem informasi jual beli tanah tersebut.



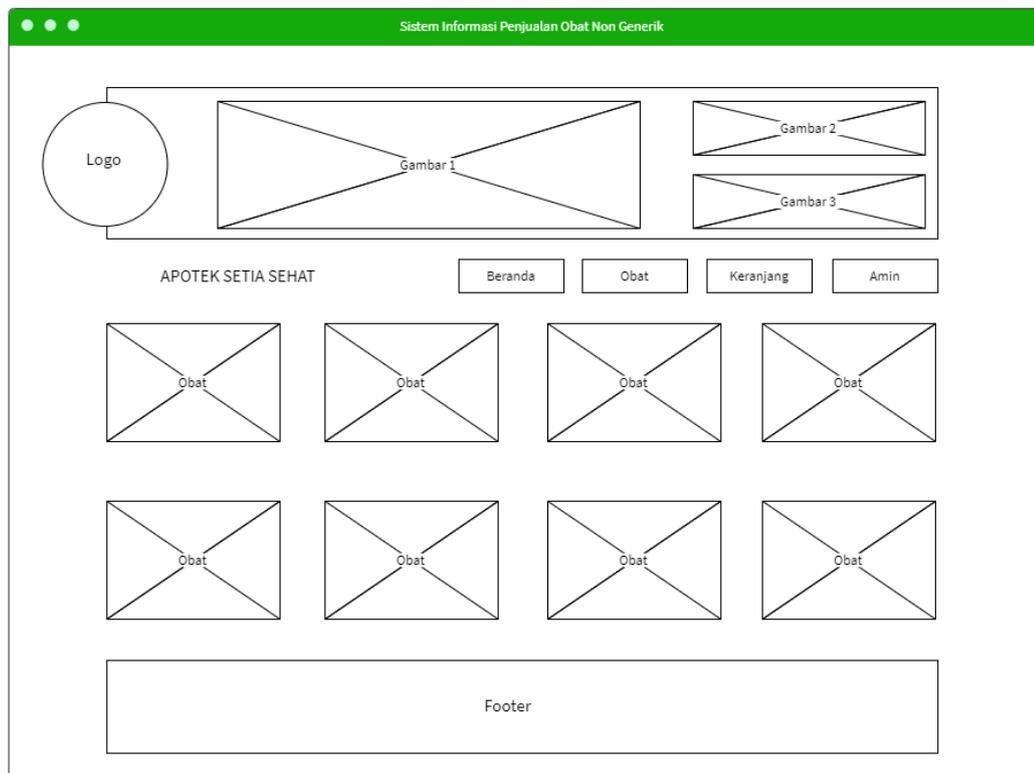
Gambar 3.5 *Flowchart* penjualan obat

3.4 Perancangan *Interface*

Interface atau yang disebut dengan antarmuka menggambarkan bagaimana bentuk *website* setelah dibangun. Perancangan ini harus dilakukan agar model dan gambaran *website* dapat dibuat dengan mudah. Antarmuka harus dapat dibangun seindah mungkin agar pengguna nyaman dalam menggunakan *website* tersebut.

3.4.1 Rancangan Beranda

Perancangan beranda merupakan halaman pertama atau *homepage* dalam sistem informasi. Gambar 3.6 merupakan hasil perancangan beranda.

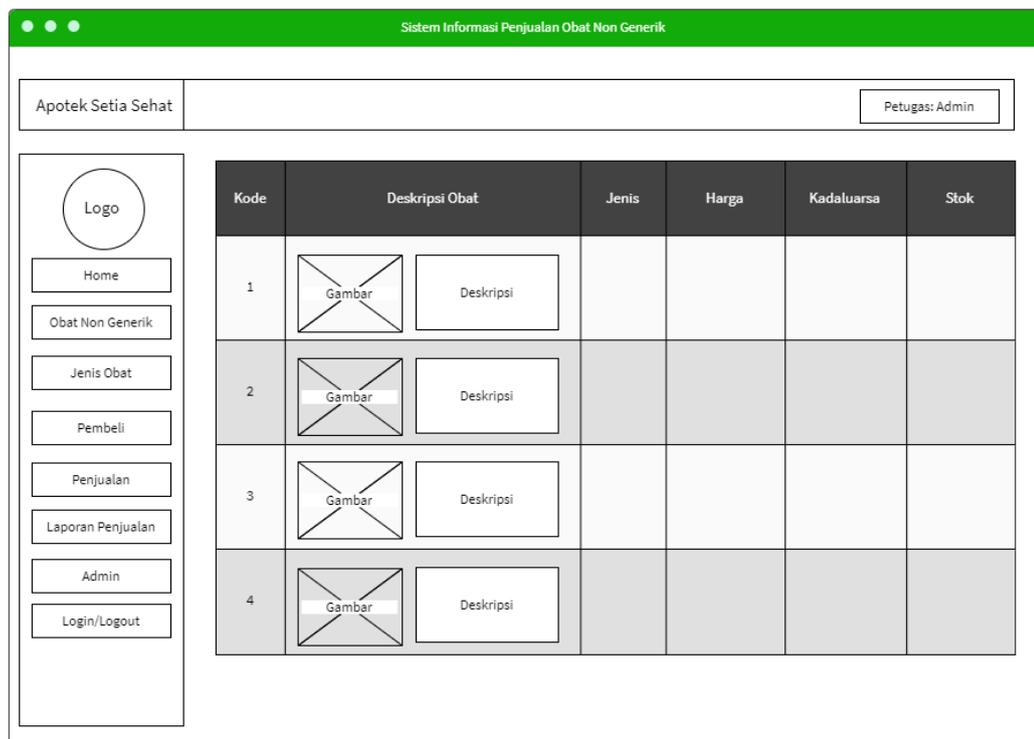


Gambar 3.6 Rancangan beranda

Gambar 3.6 menjelaskan rancangan untuk beranda. Dapat dilihat di beranda tersebut ada beberapa menu yang dapat diakses, antara lain produk, penjualan, pelanggan, dan admin. Beranda ini ditujukan kepada administrator apotek untuk dapat melakukan pengolahan data penjualan Apotek Setia Sehat.

3.4.2 Rancangan Obat Non Generik

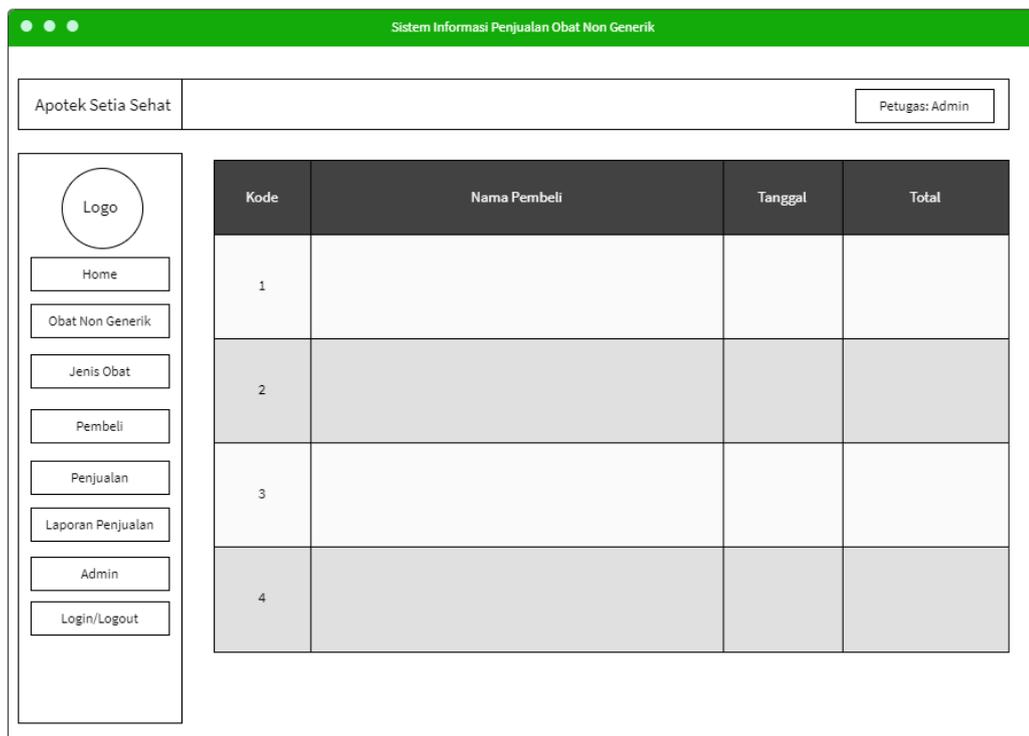
Rancangan ini menampilkan produk obat yang dijual di Apotek Setia Sehat. Rancangan ini juga menampilkan foto produk untuk mengetahui lebih jelas bentuk obat non generik. Gambar 3.7 merupakan hasil rancangan produk.



Gambar 3.7 Rancangan obat non generik

3.4.3 Rancangan Penjualan

Rancangan penjualan adalah bagian rancangan sistem yang menampilkan hasil penjualan obat yang dilakukan oleh Apotek Setia Sehat. Gambar 3.8 adalah rancangan penjualan.



Gambar 3.8 Rancangan penjualan

3.4.4 Rancangan Login

Rancangan login berfungsi untuk memberikan hak akses terbatas dalam melakukan pengelolaan data penjualan obat yang terjadi di Apotek Setia Sehat. Gambar 3.9 adalah hasil perancangan login.

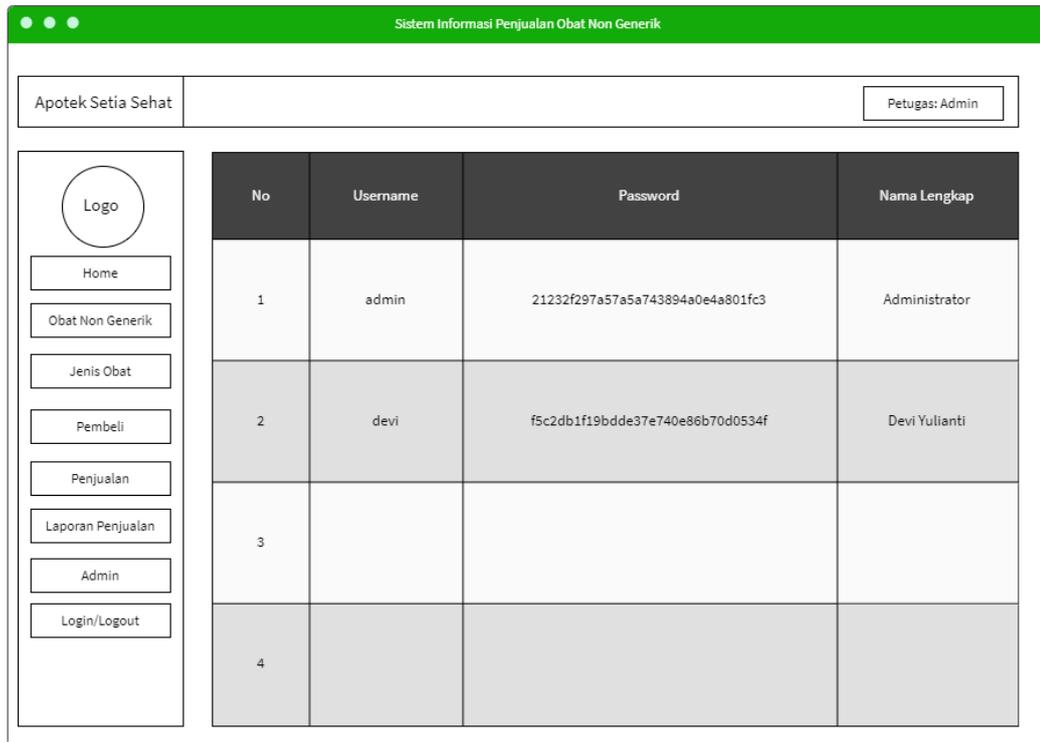


The image shows a web browser window with a green title bar containing the text "Sistem Informasi Penjualan Obat Non Generik". The main content area is white and contains a centered login form. The form has a title "Silahkan Masuk" and a subtitle "Apotek Setia Sehat". It includes three input fields: "masukkan username", "masukkan password", and a "Login" button.

Gambar 3.9 Rancangan login

3.4.5 Rancangan Admin

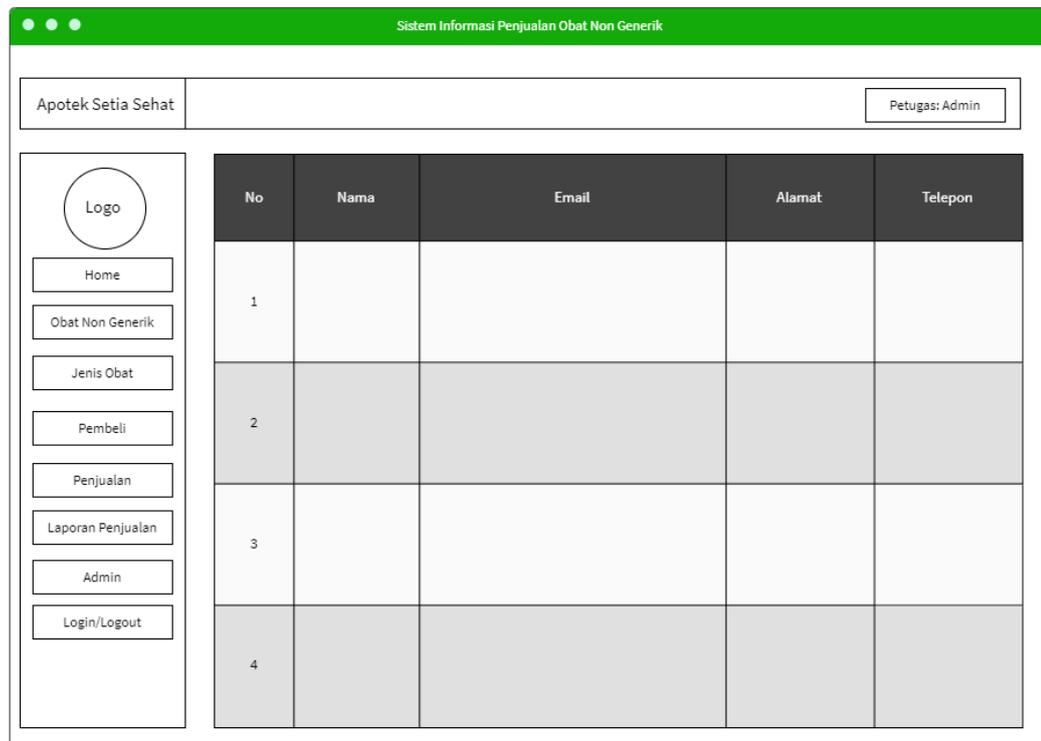
Rancangan admin memiliki beberapa bagian yang berfungsi untuk melihat pengguna yang terdaftar yang dapat mengelola data penjualan obat di Apotek Setia Sehat. Disini admin akan melakukan penambahan dan pengurangan terhadap data-data pengguna yang memiliki hak akses terhadap sistem informasi. Gambar 3.10 adalah hasil perancangan admin.



Gambar 3.10 Rancangan admin

3.4.6 Rancangan Pembeli

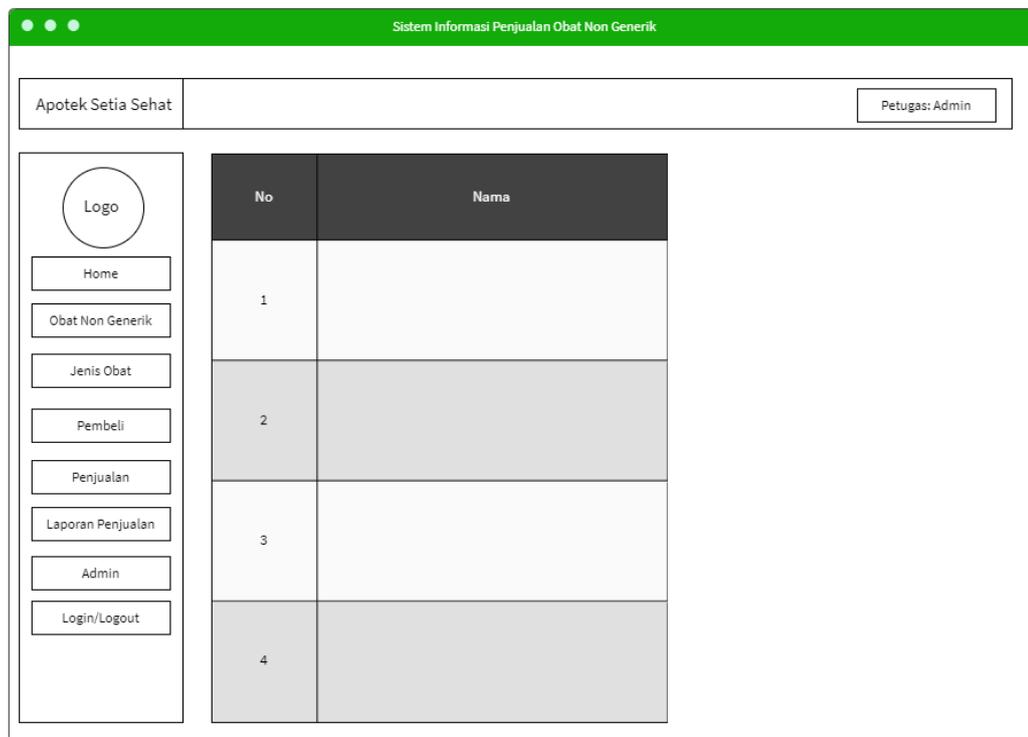
Rancangan pembeli memiliki berfungsi untuk mencatat pembeli yang sudah terdaftar dan pernah melakukan pembelian di Apotek Setia Sehat. Disini admin dapat melakukan penambahan dan pengurangan terhadap data-data pembeli yang sudah memiliki histori pembelian sebelumnya. Gambar 3.11 adalah hasil perancangan pembeli.



Gambar 3.11 Rancangan pembeli

3.4.7 Rancangan Jenis Obat

Rancangan jenis membagi kategori jenis obat yang dijual di Apotek Setia Sehat. Ada beberapa jenis yang dijual seperti *capsule*, *drop*, *syrup* dan *tablet*. Admin juga dapat melakukan penambahan dan pengurangan terhadap data-data jenis obat apabila ada penambahan jenis yang baru. Gambar 3.12 adalah hasil perancangan jenis obat.



Gambar 3.12 Rancangan jenis obat

3.5 Data Obat Non Generik

Penulis melampirkan data sampel yang digunakan untuk proses penjualan obat non generik di Apotek Setia Sehat. Ada sebanyak 30 buah data produk obat yang akan dijual yang akan dilampirkan pada sistem informasi ini. Tabel berikut adalah data produk obat non generik yang digunakan dalam penelitian ini.

Tabel 3.1 Data produk obat non generik

Nama Obat	Kandungan Obat	Fungsi Obat	Efek Samping
Fenosup Tablet	Fenofibrate 160mg	Anti Kolesterol	Sembelit, Diare, Sakit Kepala, Mual Muntah
Flamoxi Tablet	Meloxicam 15mg	Nyeri Sendi	Diare, Perut Kembang, Gangguan efek Pencernaan
Cobazim Tablet	Cobamamide	Vitamin	Perut Kembang, Mual tanpa muntah

Pidovix Tablet	Clopidogrel Bisulfate	Serangan Jantung	Diare, Perdarahan yang sulit berhenti
Tiriz Drop	Cetirizine HCl	Flu/Alergi	Mengantuk, Pusing, Lemas
Lapimuc Drop	Amboxol HCl Trimethoprim, Sulfamethoxazole	Oabt Batuk	Mengantuk, Mual dan Muntah
Sanprima Syr	Suspensi	Antibiotik	Mual, Muntah, Demam
Sporetik Syr	Cefixime Rifampicin, Isoniazid, Pyrazinamide	Antibiotik	Mual, Muntah, Demam
Pro TB 4	Nystatin	Obat Paru-paru	Sakit Kepala, Mual
Mycostatin Syr	Betamethasone, Dexchlorpheniramine Maleate	Sariawan/Jamur	Mual, Muntah, Diare
Cortamin Syr	Landson	Flu/Alergi	Mengantuk, Perut Kembang
Provital Tablet	Tranexamic Acid	Vitamin	Belum Ada Efek Samping
Nexitra Tablet	Mecobalamin	Anti Perdarahan	Pencernaan, Mual, Muntah
Methycobal Tablet	Gabapentin	Vitamin B12	Mual, Muntah, Diare
Alpentin 100mg	Pregabalin	Anti Kejang	Mual, Muntah, Kelelahan
Nomathic Tablet	NSK II 100mg, Co Q-10 30mg	Meredakan Nyeri Mengurangi Resiko Serangan Jantung	Pusing, Mengantuk
Natto Tablet	Atorvastatin Calcium	Anti Kolesterol	Belum Ada Efek Samping
Orvast 20mg	Amoxicillin	Antibiotik	Perut Sembelit
Amoxsan 500mg	Citrus Aurantium Fructus Extract	Terapi Hemoroid Akut (Wasir)	Saluran Pencernaan
Ciflon	Cal-95	Multivitamin	Belum Ada Efek Samping
Cal-95	Paracetamol	Obat Demam	Berupa Diare
Apetic Syr	Ibu Profen	Obat Demam	Mual, Muntah Mual, Muntah, Gangguan Saluran Pencernaan
Rhelafen Syr	Paracetamol	Obat Demam	Mual, Muntah
Kamolax Syr	Vitamin C	Vitamin Daya Tahan Tubuh	Pencernaan
Apialys Syr	Zinc	Vitamin Daya Tahan Tubuh	Pencernaan
Imunped Syr	Mebhydroline Napadisylate	Alergi/Gatal Pada Kulit	Pencernaan
Interhistin	Eperisone HCl	Anti Nyeri/Pegal	Mengantuk
Estalex	Isoprinodin	Obat Anti Virus	Gangguan Fungsi Hati
Isoprinodin Syr	Sericocalyx	Meluruhkan Batu Ginjal	Mengantuk
Renax Kapsul			Belum Ada Efek Samping

BAB IV

HASIL DAN PEMBAHASAN

Sistem yang dibangun akan diimplementasikan di Apotek Setia Sehat untuk mendapatkan hasil *feedback* dari orang yang menggunakan sistem informasi tersebut. Ada dua buah kategori pada sistem informasi yang dibangun yaitu *user* dan *admin*. Untuk dapat menjalankan sistem informasi dengan baik, maka dibutuhkan sistem yang memadai.

4.1 Spesifikasi Perangkat Keras

Sistem informasi yang dibangun membutuhkan perangkat keras untuk menjalankan sistem. Sebaiknya, komputer dan beberapa perangkat lain harus memadai dan mumpuni. Tabel 4.1 adalah spesifikasi perangkat keras yang digunakan dalam penelitian.

Tabel 4.1 Spesifikasi Perangkat Keras

No.	Nama Komponen	Spesifikasi
1	<i>Processor</i>	<i>Intel Core i3 1.9 GHz</i>
2	<i>RAM</i>	4 GB
3	<i>Harddisk</i>	500 GB
4	Monitor	14 <i>inch</i>

4.2 Spesifikasi Perangkat Lunak

Perangkat lunak harus mendukung dengan versi terbaru agar dapat menjalankan sistem dengan baik. Tabel 4.2 adalah spesifikasi perangkat lunak yang digunakan dalam penelitian.

Tabel 4.2 Spesifikasi Perangkat Lunak

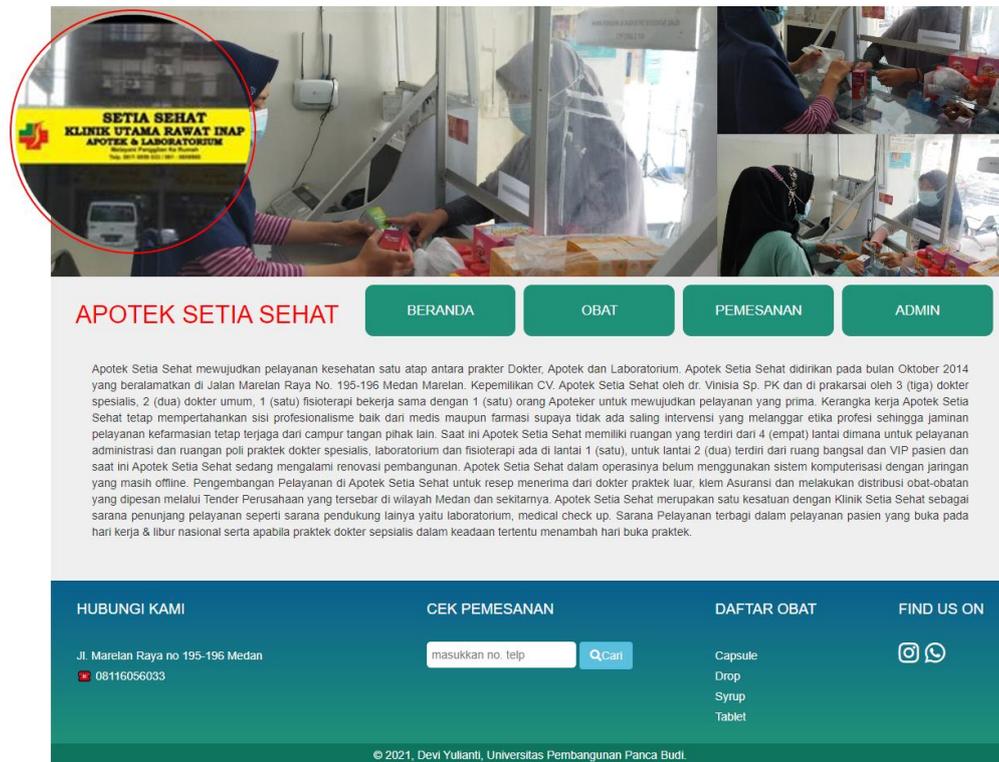
No.	Nama Komponen	Spesifikasi
1	Sistem Operasi	<i>Windows 7 32 Bit</i>
2	<i>IDE</i> Pemrograman	<i>Visual Studio Code</i>
3	<i>Webserver</i>	<i>XAMPP</i>
4	Data Editor	<i>Microsoft Excel 2019</i>
5	<i>Word Processing</i>	<i>Microsoft Word 2019</i>

4.3 Antarmuka Website

Sistem ini memiliki beberapa antarmuka pendukung dalam melaksanakan sistemnya. Ada beberapa tampilan yang dibuat pada sistem informasi ini.

4.3.1 Halaman Beranda

Halaman beranda ditujukan kepada pengguna umum yang akan membeli obat di Apotek Setia Sehat. Halaman ini menampilkan produk-produk obat yang dijual di apotek ini. Halaman ini terdiri dari beberapa menu lainnya yang mengarahkan pengguna untuk melihat isi dari menu-menu yang terdapat pada sistem informasi tersebut. Gambar 4.1 merupakan hasil tampilan beranda.



Gambar 4.1 Halaman beranda

Di halaman beranda, user dapat melakukan pemesanan obat dengan melihat stok yang ada di bawah gambar produk obat tersebut. Stok yang habis tidak dapat dipesan. Halaman beranda adalah halaman yang khusus digunakan oleh pengguna atau pembeli.

4.3.2 Halaman Obat Non Generik

Halaman obat non generik hanya ditujukan kepada pihak admin Apotik dalam melakukan perubahan data produk obat. Gambar 4.2 adalah tampilan dari halaman obat non generik.

APOTEK SETIA SEHAT PETUGAS: DEVI YULIANTI

APOTEK SETIA SEHAT

HOME

OBAT NON GENERIK

JENIS OBAT

PEMBELI

PENJUALAN

LAPORAN PENJUALAN

ADMIN

LOGOUT

TAMBAH DATA

No.	DESKRIPSI OBAT	JENIS	HARGA	KADALUARSA	STOK
1	 <p>FENOSUP KANDUNGAN: FENOFIBRATE 160MG FUNGSI: ANTI KOLESTEROL EFEK SAMPING: SEMBELIT, DIARE, SAKIT KEPALA, MUAL, MUNTAH</p> <p style="font-size: small; margin-top: 5px;">EDIT HAPUS</p>	TABLET	160.000	02 SEP 2022	8
2	 <p>FLAMOXI KANDUNGAN: MELOXICAM 15MG FUNGSI: NYERI SENDI EFEK SAMPING: DIARE, PERUT KEMBUNG, GANGGUAN EFEK PENCERNAAN</p> <p style="font-size: small; margin-top: 5px;">EDIT HAPUS</p>	TABLET	105.000	01 OCT 2022	7
3	 <p>COBAZIM KANDUNGAN: COBAMAMIDE FUNGSI: VITAMIN EFEK SAMPING: PERUT KEMBUNG, MUAL TANPA MUNTAH</p> <p style="font-size: small; margin-top: 5px;">EDIT HAPUS</p>	TABLET	25.000	20 FEB 2023	10
4	 <p>PIDOVIX KANDUNGAN: CLOPIDOGREL BISULFATE FUNGSI: SERANGAN JANTUNG EFEK SAMPING: DIARE, PERDARAHAN YANG SULIT BERHENTI</p> <p style="font-size: small; margin-top: 5px;">EDIT HAPUS</p>	TABLET	78.000	11 SEP 2024	0
5	 <p>TIRIZ KANDUNGAN: CETIRIZINE HCl FUNGSI: FLUERALERGI</p>	DROP	89.000	20 APR 2024	9

Gambar 4.2 Halaman obat non generik

Halaman ini memuat deskripsi singkat obat yang dijual yang meliputi, kandungan, fungsi dan efek samping. Admin juga dapat memperbaharui stok obat jika ada penambahan dari pemasok obat tersebut.

4.3.3 Halaman Jenis Obat

Halaman jenis obat mencatat kategori pembungkus obat atau dasar bentuk obat non generik. Ada beberapa jenis obat yang sering dijual yaitu *capsule*, *drop*, *syrup* dan *tablet*. Gambar 4.3 merupakan tampilan dari halaman jenis obat.

APOTEK SETIA SEHAT

PETUGAS: DEVI YULIANTI

TAMBAH DATA

No	NAMA	AKSI
1	CAPSULE	Edit HAPUS
2	DROP	Edit HAPUS
3	SYRUP	Edit HAPUS
4	TABLET	Edit HAPUS

Gambar 4.3 Halaman jenis obat

4.3.4 Halaman Pembeli

Halaman pembeli merupakan tampilan yang berfungsi untuk mengolah data orang yang membeli produk obat di apotek. Gambar 4.4 merupakan tampilan dari halaman pembeli.

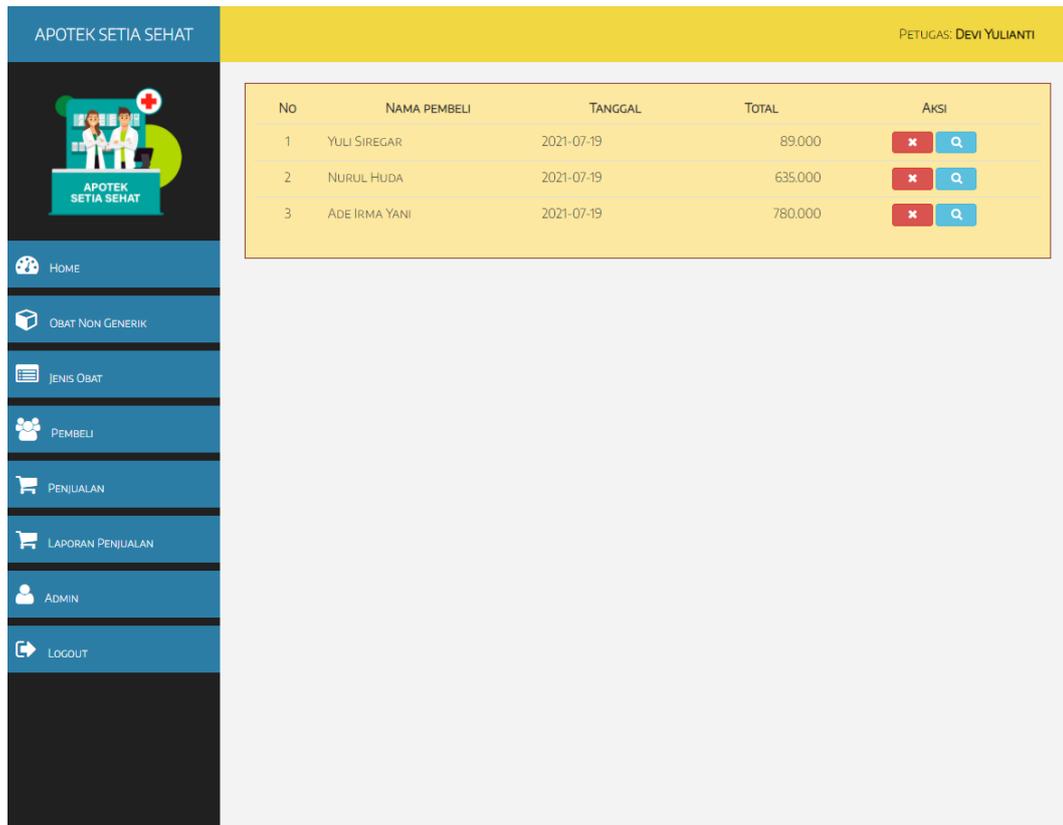
No	NAMA	EMAIL	ALAMAT	TELEPON
1	YULI SIREGAR	YULISIREGAR@GMAIL.COM	JL. SETIA BUDI PS II TPI A-49	082274997136
2	NURUL HUDA	NURUL92@YMAIL.COM	JL. MARELAN RAYA LK. 19	087889908342
3	ADE IRMA YANI	ADEIRMAYANI@GMAIL.COM	DUSUN VI PERUM BTS KUTALIMBARU	08525335742
4	DEWI RAMADHANI	DEWI94@YMAIL.COM	JL. BERSAMA CG. PERINTIS NO. 32	082267342956
5	HELENA HARTA DEWI	HELENA95@YMAIL.COM	JL. PINTU AIR IV GG. MADUMA	081269478075
6	DEVI DAMAYANTI	DEVI96@YMAIL.COM	JL. KLY SUDARSO	082276237797
7	WIDYA SAHNI	WIDYASAHNI@GMAIL.COM	JL. SETIA BUDI PSR 1	081362396107
8	SISKA YULIANDA	SISKA98@YMAIL.COM	JL. KLY SUDARSO LK. I	083166823420
9	NUR ANI LUBIS	NUR99@YMAIL.COM	Dsn. II TEGAL SARI DOLOK MASIHUL	085262178113

Gambar 4.4 Halaman pembeli

Data pembeli meliputi nama, email, alamat dan nomor telepon. Nomor telepon merupakan data yang digunakan untuk mencatat penjualan dan bersifat *unique*.

4.3.5 Halaman Penjualan

Halaman penjualan juga hanya ditujukan kepada admin dalam mencatat segala pembelian yang dilakukan oleh pengguna melalui halaman beranda. Setiap produk yang dibeli akan tercatat di penjualan dan dapat dilihat secara detail. Gambar 4.5 merupakan tampilan halaman penjualan.



No	NAMA PEMBELI	TANGGAL	TOTAL	AKSI
1	YULI SIREGAR	2021-07-19	89.000	<input type="button" value="x"/> <input type="button" value="Q"/>
2	NURUL HUDA	2021-07-19	635.000	<input type="button" value="x"/> <input type="button" value="Q"/>
3	ADE IRMA YANI	2021-07-19	780.000	<input type="button" value="x"/> <input type="button" value="Q"/>

Gambar 4.5 Halaman penjualan

Pada halaman penjualan, admin dapat melihat informasi lebih lanjut pada tombol detail yang memperlihatkan faktur pembelian dari pembeli yang sudah melakukan pembelian obat. Pada informasi detail, terdapat rincian pembelian, jumlah dan total pembayaran yang dibayarkan oleh pembeli. Tampilan ini juga dapat melakukan pencetakan ke *printer* agar pihak apotek memiliki rekap fisik terhadap penjualan. Berikut ini adalah gambar-gambar detail penjualan dan halaman pencetakan faktur.

APOTEK SETIA SEHAT

PETUGAS: DEVI YULIANTI

APOTEK SETIA SEHAT

HOME

OBAT NON GENERIK

JENIS OBAT

PEMBELI

PENJUALAN

LAPORAN PENJUALAN

ADMIN

LOGOUT

DETAIL PENJUALAN

YULI SIREGAR

No. Telp : 082274997136
E-MAIL : YULISIREGAR@GMAIL.COM

TANGGAL : 19 JUL 2021
TOTAL : 89.000

No	NAMA OBAT	JENIS	HARGA	JUMLAH	SUB TOTAL
1	TIRIZ	DROP	89.000	1	89.000
TOTAL PENJUALAN					89.000

CETAK FAKTUR

Gambar 4.6 Halaman detail penjualan

Pada halaman detail penjualan, dapat dilihat tujuan faktur tersebut. Rincian-rincian pembelian ini dapat dicetak ke *printer* dengan hanya menekan tombol *Cetak Faktur*. Selain melakukan pencetakan ke *printer*, faktur tersebut dapat disimpan dengan format *PDF*. Gambar 4.7 adalah hasil tampilan pencetakan faktur.

7/19/2021 Panel Admin Apotek Setia Sehat

DETAIL PENJUALAN

YULI SIREGAR
 NO. TELP : 082274997136
 E-MAIL : YULISIREGAR@GMAIL.COM

TANGGAL : 19 JUL 2021
 TOTAL : 89.000

NO	NAMA OBAT	JENIS	HARGA	JUMLAH	SUB TOTAL
1	TIRIZ	DROP	89.000	1	89.000
TOTAL PENJUALAN					89.000

APOTEKER
 RATNA DEWI, S.Si., Apt., M.Kes.
 445/52338/2016

localthost\SISTEM INFORMASI\SI Apotek Setia Sehat\admin\index.php?halaman=penjualan-detail&id=3 1/1

Print 1 page

Destination Save as PDF

Pages All

Layout Portrait

More settings

Save Cancel

Gambar 4.7 Halaman cetak faktur

4.3.6 Halaman Laporan Penjualan

Halaman laporan penjualan menampilkan penjualan yang terjadi dalam periode tertentu. Pengguna dapat memasukkan tanggal awal dan tanggal akhir sehingga hanya ditampilkan penjualan yang berada di rentang tanggal tersebut. Gambar 4.8 merupakan tampilan halaman laporan penjualan.

The screenshot displays the 'LAPORAN PENJUALAN' (Sales Report) page. The header includes 'APOTEK SETIA SEHAT' on the left and 'PETUGAS: DEVI YULIANTI' on the right. The sidebar on the left contains navigation items: HOME, OBAT NON GENERIK, JENIS OBAT, PEMBELI, PENJUALAN, LAPORAN PENJUALAN (highlighted), ADMIN, and LOGOUT. The main content area features a title 'LAPORAN PENJUALAN' and two date input fields: 'TANGGAL AWAL' and 'TANGGAL AKHIR', both with a placeholder 'MM/DD/YYYY' and a calendar icon. Below the date fields is a blue 'LIHAT' button. A table below the button displays sales data:

No	PEMBELI	TANGGAL	JUMLAH
1	YULI SIREGAR	2021-07-19	89.000
2	NURUL HUDA	2021-07-19	635.000
3	ADE IRMA YANI	2021-07-19	780.000
TOTAL			1504.000

Gambar 4.8 Halaman laporan penjualan

4.3.7 Halaman Admin

Halaman admin merupakan halaman yang mengelola data admin yang berhak masuk ke panel admin dari sistem informasi penjualan obat. Pihak admin dapat mengubah dan mengelola dirinya sendiri dan mengelola data admin lainnya. Gambar 4.9 merupakan tampilan halaman admin.

APOTEK SETIA SEHAT

PETUGAS: DEVI YULIANTI

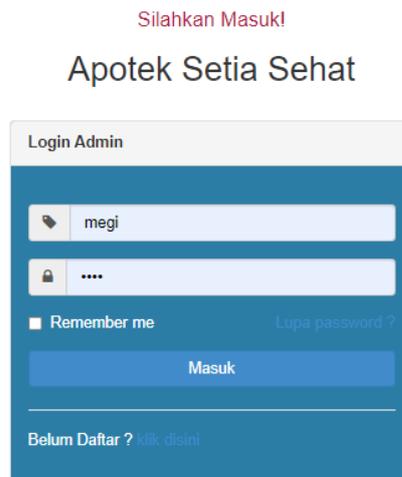
TAMBAH DATA

No	USERNAME	PASSWORD	NAMA LENGKAP	AKSI
1	ADMIN	21232f297a57a5a743894a0e4a801fc3	ADMIN TOKO	EDIT HAPUS
2	DEVI	f5c2db1f19bdde37e740e86b70d0534f	DEVI YULIANTI	EDIT HAPUS

Gambar 4.9 Halaman admin

4.3.8 Halaman *Login*

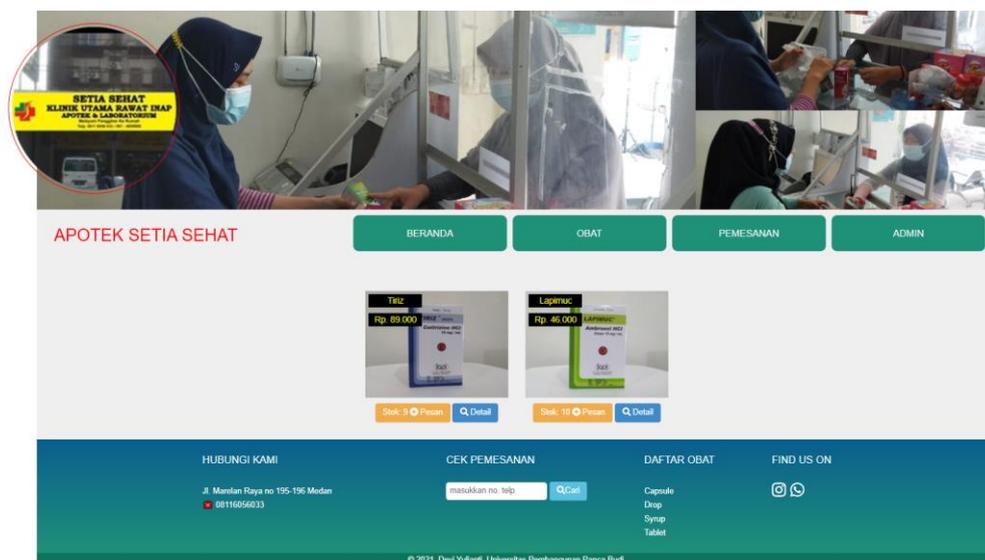
Halaman *login* berfungsi untuk membatasi akses masuk ke sistem informasi apotek bagian panel admin. Hanya pengguna yang berhak yang dapat masuk ke dalam sistem. Gambar 4.10 merupakan tampilan halaman login.



Gambar 4.10 Halaman *login*

4.3.9 Halaman Obat

Halaman login berfungsi bagi pengguna yang ingin melihat daftar obat yang dijual di apotik. Gambar 4.11 adalah halaman obat.



Gambar 4.11 Halaman obat

4.3.10 Halaman Cek Pemesanan

Halaman cek pemesanan berfungsi untuk melihat riwayat pemesanan yang dilakukan oleh pengguna. Gambar 4.12 adalah halaman cek pemesanan.

APOTEK SETIA SEHAT

BERANDA OBAT PEMESANAN ADMIN

Pemesanan Obat Non Generik

Kepada Yth:
Nurul Huda
di: Mawati Raya LK, 19
Jl. Mawati Raya
18 Jul 2021

Total Pemesanan Rp. 635.000

Nama Obat	Jumlah	Harga	Total
Fansis	2	160.000	320.000
Flamox	3	105.000	315.000

Cekah Pemesan

HUBUNGI KAMI
Jl. Mawati Raya no 195-190 Medan
08116056033

CEK PEMESANAN
masukkan no. isbn

DAFTAR OBAT
Capsule
Drop
Syring
Tablet

FIND US ON
Instagram WhatsApp

© 2021, Devi Yulianti, Universitas Pembangunan Panca Budi.

Gambar 4.12 Halaman cek pemesanan

BAB V

PENUTUP

5.1 Kesimpulan

Penelitian sistem informasi yang dibangun memberikan beberapa kesimpulan yang dapat dipaparkan, antara lain:

1. Sistem informasi dapat menjual obat non generik kepada pelanggan dengan sistem yang baik.
2. Ada beberapa tabel pada database yang tercipta yaitu tabel obat, pembeli, penjualan dan admin,
3. Hosting dilakukan secara *localhost* agar lebih memudahkan ujicoba *webiste* pada saat proses pembuatan dan pengembangan.

5.2 Saran

Penelitian ini memiliki beberapa kekurangan. Berikut ini adalah beberapa saran yang dapat dipaparkan oleh penulis, antara lain:

1. Hendaknya sistem informasi ini diterapkan secara *online*.
2. Hendaknya menambahkan daftar konsultasi pelanggan terhadap obat dan penyakit tertentu.

DAFTAR PUSTAKA

- Abdullah, D. (2015). Perancangan Sistem Informasi Pendataan Siswa SMP Islam Swasta Darul Yatama Berbasis Web. *IJNS – Indonesian Journal on Networking and Security*, 4(1), 39–44. <http://ijns.org/journal/index.php/ijns/article/view/1325/1313>
- Adisaputra, R., Hernawati, E., & Kusuma, G. P. (2018). Aplikasi Pendaftaran dan Penerimaan siswa baru berbasis web (Studi kasus: SMPN 2 CIANJUR). *Concept and Communication*, 4, 131. <https://doi.org/10.15797/concom.2019..23.009>
- Afrina, M., & Ibrahim, A. (2015). Pengembangan Sistem Informasi SMS Gateway Dalam Meningkatkan Layanan Komunikasi Sekitar Akademika Fakultas Ilmu Komputer Unsri. *Jurnal Sistem Informasi (JSI)*, 7(2), 852–86 <http://ejournal.unsri.ac.id/index.php/jsi/index>
- Aziz, A., Arry, D., & Kurnia, B. (2015). *Monitoring Serangan Pada Jaringan Komputer Menggunakan Snort Berbasis Sms Gateway*. 14(2).
- Firliana, R., Rhohman, F., & Purwinanto, R. W. (2018). Perancangan Sistem Informasi Absensi Dosen Dengan Validasi Mahasiswa Berbasis Web. *Jurnal Sains Dan Informatika*, 4(2), 105. <https://doi.org/10.34128/jsi.v4i2.142>
- Hendini, A. (2016). PEMODELAN UML SISTEM INFORMASI MONITORING PENJUALAN DAN STOK BARANG (STUDI KASUS: DISTRO ZHEZHA PONTIANAK). *Khatulistiwa Informatika*, IV(2), 107–116. <https://doi.org/10.2135/cropsoci1983.0011183x002300020002x>
- Huda Utama, E. T., Wijaya, I. D., & Hayati Ririd, A. R. T. (2017). Rancang Bangun Sistem Informasi Akademik Siswa Dengan Menggunakan Notifikasi Sms Pada Smk Muhammadiyah 1 Kepanjen. *Jurnal Informatika Polinema*, 1(4), 18. <https://doi.org/10.33795/jip.v1i4.118>
- Hutagalung, D. D., & Arif, F. (2018). Rancang Bangun Sistem Informasi Perpustakaan Berbasis Web Pada Smk Citra Negara Depok. *Journal of Chemical Information and Modeling*, 53(9), 1689–1699. <https://doi.org/10.1017/CBO9781107415324.004>
- Indrawan, M. I., Alamsyah, B., Fatmawati, I., Indira, S. S., Nita, S., Siregar, M., ... & Tarigan, A. S. P. (2019, March). UNPAB Lecturer Assessment and Performance Model based on Indonesia Science and Technology Index. In *Journal of Physics: Conference Series* (Vol. 1175, No. 1, p. 012268). IOP Publishing.
- Mufariya, Ilhamsyah, & Rahmayuda, S. (2019). *Rancang Bangun Sistem Informasi Sumbangan Pembinaan Pendidikan (Spp) Dengan Teknologi Sms Gateway*. 07(02), 19–28.
- Putra, R. R. (2019). implementasi metode backpropagation Jaringan saraf tiruan dalam memprediksi pola Pengunjung terhadap transaksi. (JurTI) *Jurnal Teknologi Informasi*, 3(1), 16-20.
- Rahim, R., Supiyandi, S., Siahaan, A. P. U., Listyorini, T., Utomo, A. P., Triyanto, W. A., ... & Khairunnisa, K. (2018, June). TOPSIS method application for decision support system in internal control for selecting best employees. In *Journal of Physics: Conference Series* (Vol. 1028, No. 1, p. 012052). IOP Publishing.

- Ruhul Amin, M. K. (2017). *Siswa Baru Pada Smk Budhi Warman 1 Jakarta*. 2(2), 113–121.
- Suendri. (2018). Implementasi Diagram UML (Unified Modelling Language) Pada Perancangan Sistem Informasi Remunerasi Dosen Dengan Database Oracle (Studi Kasus: UIN Sumatera Utara Medan). *Jurnal Ilmu Komputer Dan Informatika*, 3(1), 1–9. <http://jurnal.uinsu.ac.id/index.php/algorithm/article/download/3148/1871>
- Rusdi, M., Sirajudin, H., & Amin, M. (2020). PROTOTYPE APLIKASI PEMESANAN DAN PENGIRIMAN SERBUK KAYU OLEH CV. USAHA BERSAMA SEBAGAI SUPLIER PT. FUMAKILLA. *Technologia: Jurnal Ilmiah*, 11(3), 152-158.