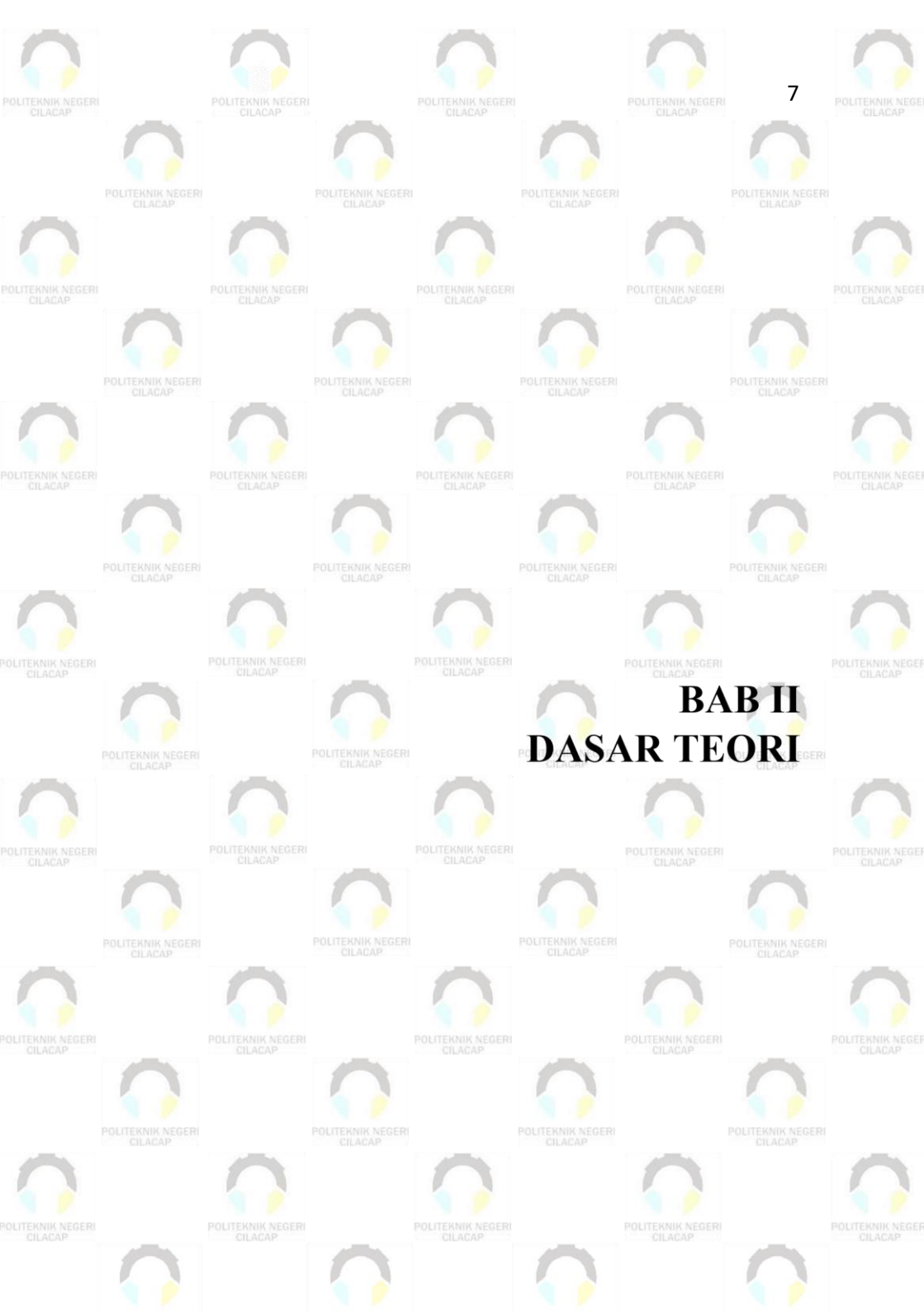


BAB II DASAR TEORI



BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Beberapa penelitian yang relevan mengenai penelitian ini pernah dilakukan oleh mahasiswa asal Universitas Islam Indonesia yaitu Sri Mulyati, Ikhawan Nur Hasyim, dan Novi Setiani. Permasalahan yang terjadi adalah bagian kemahasiswaan maupun DPA hanya mengetahui sebagian kegiatan dan prestasi dari mahasiswanya. Sistem konseling di jurusan informatika awalnya dilakukan dengan tatap muka sebelum dibuat penelitian oleh beberapa mahasiswa tersebut. Penelitian yang memuat 2 teknik; yaitu pengumpulan data, dan pengembangan sistem. Dengan berbasis website yang menggunakan bahasa pemrograman website HMTL, PHP, dan *framework* CodeIgniter[1].

Penelitian serupa juga pernah dilakukan oleh beberapa mahasiswa di Universitas Katolik Widyakarya Malang berupa Sistem Poin Mahasiswa untuk menunjang Surat Keterangan Pendamping Ijazah (SKPI). Permasalahan yang terjadi adalah pada perhitungan poin untuk memenuhi syarat SKPI masih belum ada, hal ini membutuhkan waktu yang lama dan perlu ada sistem yang mengolahnya agar lebih efisien. Penelitian Sistem e-point tersebut menggunakan framework Yii2 yang dapat menghitung secara efisien dalam proses mendapatkan sertifikat [4].

Dan penelitian terakhir yang dijadikan contoh dilakukan di Departemen Teknik Informatika, Universitas Palangka Raya.

Permasalahan yang terjadi yaitu tidak adanya sistem yang menangani atau mengolah Surat Keterangan Pendamping Ijazah di jurusan Teknik Informatika Universitas Palangka Raya. Perancangan Aplikasi Surat Keterangan Pendamping Ijazah dengan menggunakan bahasa pemrograman PHP, HTML, Bootstrap sebagai template, dan Mysql sebagai database. Sitem SKPI dapat mengolah dan mencetak deskriptif pencapaian mahasiswa seperti lomba, organisasi, atau kegiatan kemahasiswaan lainnya. Yang memuat 3 User yaitu super admin, admin, dan mahasiswa[5].

Dari beberapa penelitian diatas terdapat beberapa hal yang membedakan diantaranya ialah : Penelitian pertama oleh Sri Mulyati, Ikhawan Nur Hasyim, dan Novi Setiani dengan permasalahan pihak Akademik dan DPA hanya mengetahui sebagian kegiatan dan prestasi dari mahasiswanya. Dengan menggunakan bahasa pemrograman HTML, PHP, dan framework.

Lalu Penelitian kedua oleh mahasiswa Universitas Katolik Widyakarya Malang menggunakan framework Yii2 untuk membuat sistem poin mahasiswa untuk menunjang Surat Keterangan Pendamping Ijazah (SKPI). Dengan tujuan sistem ini dapat menghitung secara efisien dalam proses mendapatkan sertifikat.

Penelitian ketiga oleh Departemen Teknik Informatika, Universitas Palangka Raya menggunakan bahasa pemrograman PHP, HTML, Bootstrap sebagai template, dan Mysql sebagai database untuk membuat sistem informasi surat keterangan pendamping ijazah (SKPI). Tujuan sistem ini ialah dapat mengolah dan mencetak deskriptif pencapaian mahasiswa seperti lomba, organisasi, atau kegiatan

kemahasiswaan lainnya. Sistem ini memiliki tiga pengguna, yaitu super admin, admin, dan mahasiswa.

Pada sistem yang dibuat oleh penulis menggunakan bahasa pemrograman PHP, framework laravel, dan template Bootstrap serta MySQL sebagai database untuk membuat sistem informasi portofolio kegiatan mahasiswa. Sistem yang akan dibangun memiliki perbedaan fitur dari penelitian sebelumnya, yaitu dapat menyimpan data portofolio mahasiswa, menampilkan data kegiatan mahasiswa, dan mencetak data kegiatan mahasiswa. Secara keseluruhan, sistem informasi kegiatan mahasiswa merupakan pengembangan dari versi framework yang berbeda, fitur tampilan dengan versi yang lebih baru.

2.2 Landasan Teori

2.2.1. Portofolio

Portofolio adalah kumpulan dokumen yang berasal dari pribadi, kelompok, organisasi, lembaga, perusahaan dll yang tersusun rapi dari semua pekerjaan yang telah dilakukan[6].

2.2.2. Website

Website adalah suatu media yang terdiri dari beberapa halaman yang saling berkaitan satu sama lain, dan berfungsi sebagai media untuk menampilkan suatu informasi, baik berbentuk gambar, video, teks, suara, ataupun gabungan dari semuanya. *Website* bersifat *multiplatform* yang artinya dapat dibuka dari segala perangkat atau *device* yang terhubung dengan jaringan internet[7].

2.2.3. Basis Data (Database)

Database adalah kumpulan data yang yang didefinisikan dan dikelola dengan baik. Fungsi *database* adalah untuk mengelompokkan data, pengolahan informasi, menghindari duplikasi dan data yang tidak konsisten[8].

Berikut Merupakan contoh-contoh dari database :

a. DBMS (*Database Management System*)

DBMS (*Database Management System*) merupakan perantara dari *user* dengan basis data. Dalam penggunaannya memerlukan bahasa basis data yang sudah ditetapkan dari perusahaan DBMS, terdiri dari berbagai macam intruksi yang diformulasikan sehingga dapat diproses oleh DBMS.

b. SQL (*Structured Query Language*)



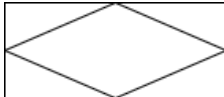



Structured Query Language (SQL) dibangun di laboratorium IBM-san Jose California sekitar tahun 70-an, merupakan bahasa yang banyak digunakan dalam berbagai produk *database* yang pertama kali dirancang sebagai bahasa DB2 yang menjadi produk andalan IBM. Saat ini (ANSI) organisasi Standar Amerika menetapkan standar bahasa SQL yaitu ANSI-92 standard.

c. ERD (*Entity Relationship Diagram*)

Entity Relationship Diagram (ERD) adalah teknik pendekatan yang menyatakan atau menggambarkan hubungan suatu model, hubungan tersebut dinyatakan untuk menunjukkan objek data *entity* dan hubungan *relationship* yang ada pada *entity* berikutnya. Hal itu dapat memungkinkan seorang analis menghasilkan struktur basis data yang

dapat diambil dan disimpan secara efisien. Berikut Tabel 2.1 adalah simbol-simbol dalam ERD :

Tabel 2. 1 *Simbol-simbol dalam ERD*





No	Simbol	Nama	Keterangan
1.		<i>Entity</i>	Suatu entitas yang memiliki nama benda, orang atau lokasi digambarkan dengan persegi panjang
2.		<i>Weak Entity</i>	Entitas yang tidak dapat diidentifikasi melalui atributnya sendiri. Weak entity bergantung pada enty lain (<i>owner entity</i>)
3.		<i>Associative</i>	Entitas yang digunakan pada banyak antar banyak hubungan
4.		<i>Attribute</i>	Mendeskripsikn karakter entitas
5.		<i>Key attribute</i>	Atribut yang berfungsi sebagai kunci
6.		<i>Link</i>	Penghubung antar relasi dengan entitas dan entitas dengan atribut.


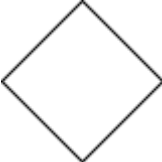
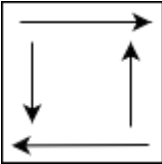
2.2.4. Flowchart


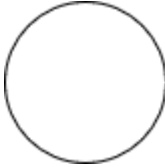
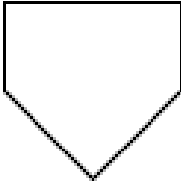

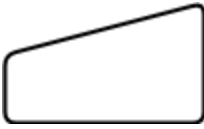
Flowchart merupakan penggambaran secara grafik dari langkah langkah dan urutan prosedur suatu program,. Biasanya mempengaruhi penyelesaian masalah yang khususnya perlu dipelajari dan dievaluasi lebih lanjut[9].

Berikut Tabel 2.2 adalah simbol-simbol dalam flowchart :

Tabel 2. 2 Simbol-simbol dalam *Flowchart*

No	Simbol	Nama	Keterangan
1.		<i>Terminal</i>	Mulai dan mengakhiri suatu program
2.		<i>Proses</i>	Proses perhitungan atau proses pengolahan data
3.		<i>Predefined Process (sub program)</i>	Permulaan sub program atau proses pengolahan data
4.		<i>Preparation</i>	Proses inialisasi atau pemberian harga awal

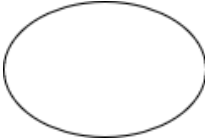

5.		<i>Input-Output</i>	Memasukan data maupun menunjukkan hasil dari suatu <i>process</i> tanpa tergantung dengan jenis peralatanya
6.		<i>Decision</i>	Memilih proses berdasarkan kondisi yang ada
7.		<i>Flow</i>	Menghubungkan antara simbol satu dengan simbol yang lain atau menyatakan jalannya arus dalam suatu proses. Simbol arus ini sering disebut dengan <i>connecting Line</i>



8.		<i>Document</i>	Merupakan simbol untuk data yang terbentuk informasi
9.		<i>On Page Connector</i>	Penghubungan bagian-bagian <i>flowchart</i> yang berada pada suatu halaman
10.		<i>Off Page Connector</i>	Penghubung bagian-bagian <i>flowchart</i> yang berada pada halaman berbeda
11.		<i>Manual Operation</i>	Menunjukkan pengolahan yang tidak dilakukan oleh komputer
12.		<i>Manual Input</i>	Memasukan data secara manual <i>on-line keyboard</i>

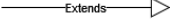
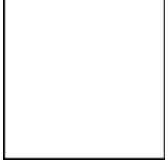

2.2.5. Use Case

Use case diagram merupakan diagram yang menjelaskan manfaat sistem dari sudut pandang orang yang berada di luar sistem atau *actor*[10]. Berikut tabel 2.3 merupakan simbol-simbol yang ada pada *use case* diagram :

Tabel 2. 3 Simbol-simbol dalam *Use Case Diagram*

No	Simbol	Nama	Keterangan
1.		<i>Terminator</i>	Fungsionalitas yang disediakan sistem sebagai unit atau aktor yang saling bertukar pesan dinyatakan dengan menggunakan kata kerja.
2.		<i>Actor</i>	Peran yang berinteraksi dengan sistem informasi yang dibuat, simbol aktor berupa gambar orang tetapi aktor belum tentu merupakan orang,

			biasanya dinyatakan dengan kata benda.
3.		<i>Association</i>	Komunikasi antara aktor dan <i>use case</i> , menghubungkan objek satu dengan yang lainnya.
4.		<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.




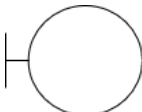
5.		<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> yang dapat berdiri sendiri walau tanpa <i>use case</i> tambahan.
6.		<i>System</i>	Paket yang menampilkan sistem secara terbatas.
7.		<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.


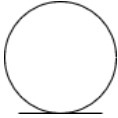
2.2.6. Sequence Diagram

Sequence diagram digunakan untuk membantu dalam memahami persyaratan sistem baru, mendokumentasikan proses dan memvisualisasikan skenario teknis saat sistem sedang dijalankan (runtime), sehingga pengguna dapat memahami dan memprediksi

bagaimana suatu sistem akan berperilaku[10]. Berikut Tabel 2.4 adalah simbol-simbol dalam *Sequence Diagram* :

Tabel 2. 4 Simbol-simbol dalam *Sequence Diagram*

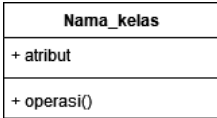
No	Simbol	Nama	Keterangan
1.		<i>Lifeline</i>	Objek <i>entity</i> , yang saling berinteraksi.
2.		<i>Actor</i>	Mengambarkan <i>user</i> atau pengguna.
3.		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi.
4.		<i>Boundary</i>	Mengambarkan sebuah <i>form</i> .



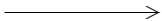


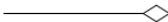
5.		<i>Control Class</i>	Menghubungkan <i>boundary</i> dengan tabel.
6.		<i>Entity Class</i>	Mengambarkan hubungan kegiatan yang akan dilakukan.

2.2.7. Class Diagram

Class diagram atau diagram kelas adalah salah satu jenis diagram struktur pada UML yang menggambarkan dengan jelas struktur serta deskripsi class, atribut, metode, dan hubungan dari setiap objek. Ia bersifat statis, dalam artian diagram kelas bukan menjelaskan apa yang terjadi jika kelaskelasnya berhubungan, melainkan menjelaskan hubungan apa yang terjadi[11]. Berikut Tabel 2.3 adalah simbol-simbol yang terdapat pada *class* diagram :

Tabel 2. 5 Simbol-simbol dalam *Class Diagram*

No	Simbol	Nama	Keterangan
1.		Kelas	Kelas pada struktur sistem

2.		Antarmuka/ <i>interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3.		Asosiasi/ <i>association</i>	Relasi antar kelas yang biasanya disertai dengan <i>multiplicity</i> .
4.		Asosiasi berarah/ <i>directed association</i>	Relasi antar kelas yang satu digunakan oleh kelas yang lain, biasanya disertai dengan <i>multiplicity</i> .
5.		Generalisasi	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6.		Kebergantungan/ <i>dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas
7.		Agregasi/ <i>aggregation</i>	Relasi antar kelas dengan makna semua bagian.

2.2.8. Pengujian Sistem

Pengujian sebuah sistem merupakan salah satu hal yang penting untuk dilakukan dengan tujuan untuk memeriksa kesalahan – kesalahan yang terjadi didalam sistem yang dibangun dan meminimalisir terjadinya kerugian dari kesalahan-kesalahan yang mungkin terjadi di sistem[12]. Berikut adalah beberapa jenis pengujian sistem yang umum dilakukan dalam pengembangan perangkat lunak:

- a. Pengujian Fungsional: Melibatkan pengujian komponen atau fitur sistem untuk memastikan bahwa sistem berperilaku sesuai dengan persyaratan fungsional yang telah ditentukan.
- b. Pengujian Non-Fungsional: Melibatkan pengujian aspek non-fungsional sistem seperti kinerja, keamanan, skalabilitas, ketersediaan, dan kegunaan.
- c. Pengujian Integrasi: Dilakukan untuk menguji interaksi antara komponen atau modul sistem yang berbeda setelah diintegrasikan. Hal ini bertujuan untuk memastikan bahwa komponen-komponen tersebut dapat bekerja bersama dengan baik.
- d. Pengujian Sistem: Pengujian sistem dilakukan pada tingkat keseluruhan sistem untuk memeriksa apakah sistem dapat beroperasi dengan benar dalam lingkungan yang direncanakan.
- e. Pengujian Unit: Melibatkan pengujian setiap unit atau komponen perangkat lunak secara terisolasi untuk memastikan bahwa unit tersebut berfungsi dengan benar secara independen.

- f. Pengujian Regresi: Pengujian regresi dilakukan untuk memastikan bahwa perubahan atau pembaruan yang dilakukan pada sistem tidak mempengaruhi fungsi yang sudah ada sebelumnya.
- g. Pengujian Aplikasi Web: Melibatkan pengujian fungsi dan kinerja aplikasi web seperti kecocokan antara tampilan antarmuka pengguna dengan berbagai browser dan perangkat yang berbeda.
- h. Pengujian Keamanan: Melibatkan pengujian sistem untuk mengidentifikasi kerentanan keamanan yang mungkin ada dan melindungi sistem dari ancaman yang tidak diinginkan.
- i. Pengujian Beban: Melibatkan pengujian kinerja sistem dengan memberikan beban yang tinggi untuk menguji batasan dan kapasitas sistem.
- j. Pengujian Penerimaan Pengguna: Dilakukan oleh pengguna akhir atau pemangku kepentingan untuk memastikan bahwa sistem memenuhi persyaratan bisnis dan harapan mereka.

2.2.9. Framework

Framework adalah koleksi atau kumpulan potongan-potongan program yang disusun atau diorganisasikan sedemikian rupa, sehingga dapat digunakan untuk membantu membuat aplikasi utuh tanpa harus membuat semua kodenya dari awal [13].

2.2.10. Laravel

Laravel adalah sebuah MVC web development framework yang didesain untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan dan perbaikan serta meningkatkan produktifitas pekerjaan dengan sintak yang bersih dan fungsional yang

dapat mengurangi banyak waktu untuk implementasi. Laravel merupakan framework dengan versi PHP yang up-to-date, karena Laravel mensyaratkan PHP versi 5.3 keatas. Laravel merupakan framework PHP yang menekankan pada kesederhanaan dan fleksibilitas pada desainnya.

Laravel memberikan keterbaruan alat untuk berinteraksi dengan database disebut dengan migration. Dengan migration, pengembang dapat dengan mudah untuk melakukan modifikasi sebuah database pada sebuah platform secara independen karena implementasi skema database direpresentasikan dalam sebuah class. Migration dapat berjalan pada beberapa basis data yang telah didukung Laravel (MySQL, PostgreSQL, MSSQL, dan SQLITE) dan untuk implementasi Active Record pada Laravel disebut Eloquent yang menggunakan standard modern OOP. Laravel juga memberikan sebuah Command Line Interface disebut dengan artisan, pengembang dapat berinteraksi dengan aplikasi untuk sebuah aksi seperti migrations, testing, atau membuat controller dan model. Selain itu, laravel juga memiliki Blade template engine yang memberikan estetika dan kebersihan kode pada view secara parsial[14].

2.2.11.Sistem Informasi

Sistem Informasi adalah kumpulan sub-subsistem yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan, yaitu mengelola data menjadi informasi yang berguna [15]. Ada beberapa jenis sistem informasi yang digunakan dalam berbagai bidang dan organisasi. Berikut adalah beberapa contoh jenis-jenis sistem informasi yang umum:

Sistem Informasi Manajemen (SIM): Sistem Informasi Manajemen membantu dalam pengumpulan, penyimpanan, pengelolaan, dan pemrosesan data untuk mendukung pengambilan keputusan manajerial. SIM membantu dalam mengintegrasikan data dari berbagai departemen atau fungsi dalam organisasi, seperti keuangan, sumber daya manusia, penjualan, produksi, dan lainnya.

- a. Sistem Informasi Eksekutif (SIE): Sistem Informasi Eksekutif dirancang khusus untuk memenuhi kebutuhan informasi tingkat eksekutif dalam organisasi. Ini memberikan informasi yang relevan, cepat, dan ringkas kepada para eksekutif untuk membantu mereka dalam pengambilan keputusan strategis.
- b. Sistem Pendukung Keputusan (SPK): Sistem Pendukung Keputusan membantu para pengambil keputusan dalam mengatasi masalah kompleks dengan menyediakan informasi, model, dan alat analisis yang mendukung proses pengambilan keputusan. SPK menggunakan teknik-teknik seperti pemodelan, analisis data, dan algoritma untuk memberikan saran dan rekomendasi kepada pengguna.
- c. Sistem Informasi Geografis (SIG): Sistem Informasi Geografis digunakan untuk mengumpulkan, mengelola, menganalisis, dan memvisualisasikan data yang berhubungan dengan lokasi geografis. SIG menggabungkan data geografis dengan atribut non-geografis untuk membantu dalam pemetaan, perencanaan wilayah, pemantauan lingkungan, navigasi, dan pengambilan keputusan yang berkaitan dengan lokasi.
- d. Sistem Informasi Pemasaran (SIP): Sistem Informasi Pemasaran membantu dalam pengumpulan, analisis, dan penggunaan data yang

- berkaitan dengan kegiatan pemasaran, seperti penjualan, promosi, pelanggan, dan penelitian pasar. SIP membantu dalam memahami pasar, pelanggan, tren, dan pola perilaku konsumen untuk mengembangkan strategi pemasaran yang efektif.
- e. Sistem Informasi Keuangan (SIK): Sistem Informasi Keuangan digunakan untuk mengelola data dan proses yang berkaitan dengan keuangan dan akuntansi dalam organisasi. SIK membantu dalam mengelola pembayaran, pembukuan, pelaporan keuangan, perencanaan anggaran, dan audit keuangan.
 - f. Sistem Informasi Sumber Daya Manusia (SISDM): Sistem Informasi Sumber Daya Manusia membantu dalam manajemen data dan proses yang berkaitan dengan sumber daya manusia dalam organisasi. Ini mencakup pengelolaan data karyawan, rekrutmen, pelatihan, penggajian, kinerja, dan manajemen talenta.
 - g. Sistem Informasi Logistik (SIL): Sistem Informasi Logistik digunakan untuk mengelola aliran barang dan informasi terkait dalam rantai pasokan. SIL membantu dalam perencanaan persediaan, pengelolaan persediaan, pengiriman, pelacakan pengiriman, dan manajemen pergudangan.

2.2.12. Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah salah satu alat bantu atau pemodelan yang sangat handal di dunia pengembangan sistem yang berorientasi obyek. UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh Booch, *Object Oriented Modeling Technique (OMT)* dan *Object Oriented Software Engineering (OOSE)*.

UML terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Yaitu :

1. Diagram Struktur (Structural Diagrams):

- a. *Class Diagram*
- b. *Object Diagram*
- c. *Component Diagram*
- d. *Deployment Diagram*
- e. *Package Diagram*
- f. *Composite Structure Diagram*

2. Diagram Perilaku (*Behavioral Diagrams*):

- a. *Use Case Diagram*
- b. *Activity Diagram*
- c. *Sequence Diagram*
- d. *Communication Diagram* (sebelumnya disebut *Collaboration Diagram*)
- e. *State Machine Diagram*
- f. *Timing Diagram*

3. Diagram Interaksi (*Interaction Diagrams*):

- a. *Sequence Diagram*
- b. *Communication Diagram* (sebelumnya disebut *Collaboration Diagram*)
- c. *Timing Diagram*