

BAB II TINJAUAN PUSTAKA

2.1 Sistem Informasi

Pengertian menurut Krismiaji (2015) Sistem informasi adalah cara-cara yang diorganisasi untuk mengumpulkan, memasukkan, dan mengolah serta menyimpan data, dan cara-cara yang diorganisasi untuk menyimpan, mengelola, mengendalikan, dan melaporkan informasi sedemikian rupa sehingga sebuah organisasi dapat mencapai tujuan yang telah ditetapkan.

Definisi menurut Nas (2018) Sistem informasi adalah sistem pemrosesan data, merupakan sistem buatan manusia yang biasanya terdiri dari sekumpulan komponen (baik manual maupun berbasis komputer) yang terintegrasi untuk mengumpulkan, menyimpan, dan mengelola data serta menyediakan informasi mengenai saldo persediaan.

Jadi berdasarkan pengertian diatas dapat disimpulkan bahwa sistem informasi adalah kumpulan data yang terintegritasi dan saling melengkapi dengan menghasilkan *output* yang baik guna untuk memecahkan masalah dan pengambilan keputusan.

2.2 Sistem Informasi Administrasi Kependudukan

SIAK adalah Sistem Informasi Administrasi Kependudukan, yaitu suatu sistem informasi yang disusun berdasarkan prosedur-prosedur dan memakai standarisasi khusus yang bertujuan menata sistem administrasi kependudukan sehingga tercapai tertib administrasi di bidang kependudukan. Administrasi kependudukan meliputi Pendaftaran Penduduk dan Pencatatan Sipil [9].

Definisi Sistem Informasi Administrasi Kependudukan (SIAK) berdasarkan Undang-Undang Nomor 23 Tahun 2006 tentang Administrasi Kependudukan adalah rangkaian kegiatan penataan dan penertiban dalam penerbitan dokumen dan Data Kependudukan melalui Pendaftaran Penduduk, Pencatatan Sipil,

pengelolaan informasi Administrasi Kependudukan serta pendayagunaan hasilnya untuk pelayanan publik dan pembangunan sektor lain.

Menurut Peraturan Pemerintah Nomor 37 Tahun 2007 Tentang Pelaksanaan Undang-Undang Nomor 23 Tahun 2006 Tentang Administrasi Kependudukan, Pengelolaan SIAK bertujuan:

- a. Meningkatkan kualitas pelayanan Pendaftaran Penduduk dan Pencatatan Sipil.
- b. menyediakan data dan informasi skala nasional dan daerah mengenai hasil pendaftaran penduduk dan pencatatan sipil yang akurat, lengkap, mutakhir dan mudah diakses.
- c. mewujudkan pertukaran data secara sistemik melalui sistem pengenalan tunggal, dengan tetap menjamin kerahasiaan.

Sistem Informasi Administrasi Kependudukan mempunyai peranan antara lain:

- a. Perekaman, pengiriman dan pengolahan data hasil pendaftaran penduduk dan pencatatan sipil.
- b. Penerbitan NIK Nasional.
- c. Memfasilitasi validasi dan verifikasi individu pendudukan untuk pelayanan publik lainnya.
- d. Penyajian data dan informasi yang mutakhir bagi instansi terkait dalam rangka perencanaan pembangunan dan pelaksanaan program pemerintah.

2.3 Profil Kependudukan

Profil Kependudukan berisi gambaran kondisi, perkembangan dan prospek kependudukan suatu daerah yang diharapkan dapat memberikan informasi, pendidikan, jenis pekerjaan, penghasilan, umur, rasio jenis kelamin serta penyediaan sarana dan prasarana yang berkaitan dengan pembangunan kependudukan [9].

Keputusan Menteri Dalam Negeri No 54 tentang Kependudukan Pedoman Penyelenggaraan Pendaftaran Penduduk yang dijelaskan diantaranya:

- a. Penduduk, adalah Warga Negara Indonesia (WNI) dan Warga Negara Asing (WNA) pemegang ijin tinggal tetap di wilayah Negara Kesatuan Republik Indonesia atau semua orang yang berdomisili di desa tersebut selama enam

- bulan lebih atau mereka yang berdomisili kurang dari enam bulan tetapi bertujuan menetap.
- b. Keluarga, merupakan unit terkecil dalam masyarakat yang mempunyai hubungan darah dan orang lain yang tinggal dalam satu rumah atau bangunan yang terdaftar dalam kartu keluarga.
 - c. Kepala Keluarga, laki-laki atau perempuan yang berstatus kawin, janda atau duda yang mengepalai satu keluarga yang anggotanya terdiri dari istri atau suami dan anak-anak.
 - d. Anggota Keluarga, mereka yang tercantum dalam satu kartu keluarga dua orang atau lebih dan salah satu ada yang menjadi kepala keluarga.

2.4 CodeIgniter

CodeIgniter menurut [11] merupakan *framework* untuk bahasa pemrograman PHP, yang dibuat Rick Ellis pada tahun 2006. *CodeIgniter* memiliki banyak fitur yang membantu para pengembang PHP untuk dapat membuat aplikasi secara mudah dan cepat serta memiliki sifat yang fleksibel dapat dikembangkan dalam perangkat *web*, dekstop maupun *mobile*. *CodeIgniter* memiliki konsep atau pola *Model-View-Controller* (MVC) sehingga kode-kode dapat di sederhanakan.

2.4.1 MySql

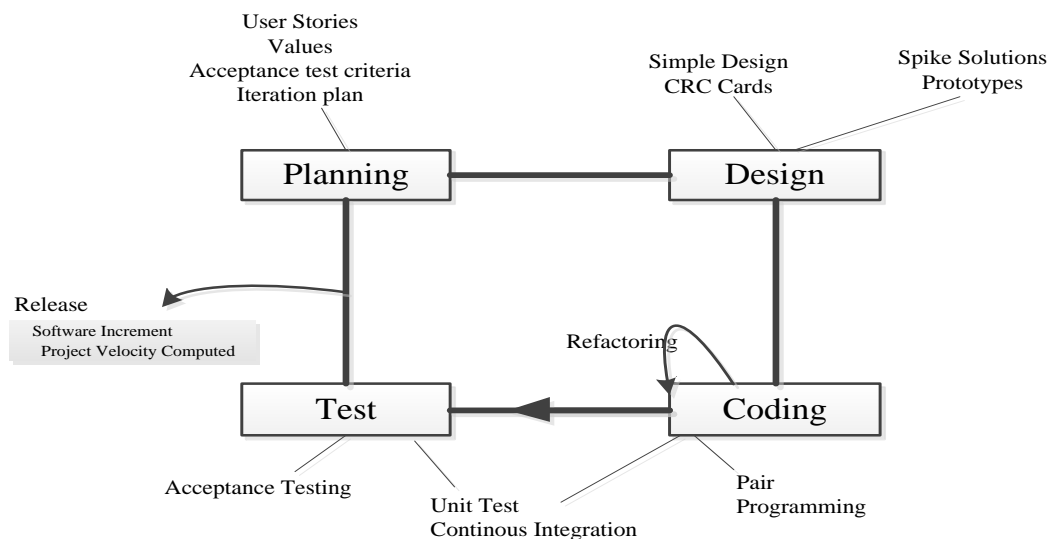
MySQL Sabar, Heryanto dan Lestari (2019) merupakan basis data yang bersifat open source sehingga banyak di gunakan untuk media. Walaupun gratis, MySQL tetap berkualitas dan sudah cukup memberikan performance yang memadai. Penggunaan PHP MyAdmin lebih mudah digunakan karena menggunakan interface yang lebih mudah dipahami.

Menurut [13], MySQL adalah nama sebuah database server yang menangani akses database yang selalu dalam bentuk pernyataan SQL (*Structured Query Language*) yaitu suatu bahasa yang digunakan untuk mengakses database relasional.

2.5 Extreme Programming

Menurut Suryantara (2017) *extreme programming* merupakan salah satu metodologi rekayasa perangkat lunak yang banyak digunakan untuk mengembangkan aplikasi oleh para developer dan merupakan pengembangan proyek yang memerlukan adaptasi yang cepat dalam perubahan-perubahan yang terjadi selama pengembangan aplikasi. Pengembangan Extreme Programming (XP) karena dinilai sangat efektif dalam perancangan suatu sistem informasi [15].

Tujuan penggunaan metode tersebut untuk menghasilkan perangkat lunak yang berkualitas serta produktif dengan siklus pengembangan yang cukup singkat. Berikut merupakan konsep *extreme programming* yang dapat dilihat pada Gambar 2.1.



Gambar 2.1 *Extreme Programming*

Sumber: (Suryantara, 2017)

a. Kelebihan Model Extreme Programming

Komunikasi dalam XP dibangun dengan melakukan pemrograman berpasangan (*pair programming*). *Developer* didampingi oleh pihak klien dalam melakukan *coding* dan *unit testing* sehingga klien bisa terlibat langsung dalam pemrograman sambil berkomunikasi dengan *developer*. Selain itu perkiraan beban tugas juga diperhitungkan.

1. Menekankan pada kesederhanaan dalam pengkodean: “*What is the simplest thing that could possibly work?*” Lebih baik melakukan hal yang sederhana dan mengembangkannya besok jika diperlukan. Komunikasi yang lebih banyak mempermudah, dan rancangan yang sederhana mengurangi penjelasan.
2. Setiap *feed back* ditanggapi dengan melakukan tes, *unit test* atau *system integration* dan jangan menunda karena biaya akan membengkak (uang, tenaga, waktu).
3. Banyak ide baru dan berani mencobanya, berani mengerjakan kembali dan setiap kali kesalahan ditemukan, langsung diperbaiki.

b. Kelemahan Model Extreme Programming

Kelemahan pada pengembangan tersebut dapat dilihat berdasarkan kesesuaian pengembangan yang dilakukan, berikut adalah kelemahan metode pengembang sistem:

1. *Developer* harus selalu siap dengan perubahan karena perubahan akan selalu diterima.
2. Tidak bisa membuat kode yang detail di awal (*prinsip simplicity* dan juga anjuran untuk melakukan apa yang diperlukan hari itu juga).

2.6 Alat Pengembang Sistem (*Unified Modelling Language*)

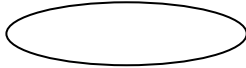
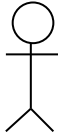

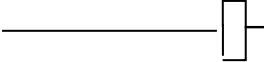
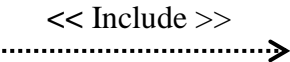
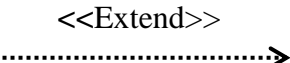
Alat pengembang sistem merupakan konsep desain yang digunakan untuk menggambarkan sistem dengan menggunakan diagram. Penyesuaian alat yang digunakan harus sesuai dengan metode pengembangan yang dilakukan salah satunya adalah penerapan *Unified Modelling Language*. Menurut (Rosa and Shalahuddin, 2019) UML (*unified Modelling Language*) adalah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. Berikut ini merupakan penjelasan tentang masing-masing diagram yang ada pada UML (*Unified Modelling Language*).

2.6.1 *Use Case Diagram*

Menurut Rosa and Shalahuddin (2019) *Use Case* adalah *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem

informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Use Case Diagram* dapat dilihat pada tabel 2.1.

Tabel 2.1 Simbol *Use Case Diagram*



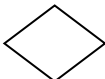

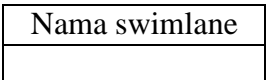

No	Simbol	Deskripsi
1.		Usecase Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal <i>frase</i> nama <i>use case</i> .
2.		Aktor Aktor seseorang/sesuatu yang berinteraksi dengan yang akan dibuat. diluar sistem informasi. Biasanya dinyatakan menggunakan kata benda
3.		Asosiasi/association merupakan komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.
4.		Generalisasi (<i>generalization</i>) merupakan hubungan (umum – khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum
5.		Include berarti use case yang ditambahkan akan dipanggil saat use case tambahan dijalankan.
6.		Ekstensi (<i>extend</i>) merupakan use case tambahan ke sebuah use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu.

Sumber : (Rosa and Shalahuddin, 2019)

2.6.2 *Activity Diagram*

Menurut Rosa and Shalahuddin (2019) *activity diagram* adalah *activity Diagram* menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *activity diagram* dapat dilihat pada tabel 2.3 berikut ini :

Tabel 2.2 Simbol *Activity Diagram*

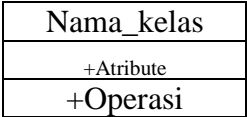


No.	Simbol	Keterangan
1.		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.		Percabangan (<i>Decision</i>) merupakan asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.		Penggabungan (<i>Join</i>) merupakan asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.		Swimlane Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas.
6.		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

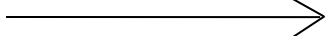
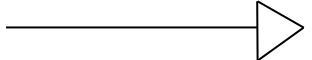
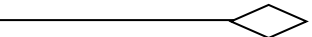
Sumber : (Rosa and Shalahuddin, 2019)

2.6.3 *Class Diagram*

Menurut Rosa and Shalahuddin (2019) *Class Diagram* adalah *Class diagram* mengembangkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Class Diagram* dapat dilihat pada tabel 2.3 berikut ini :

Tabel 2.3 Simbol *Class Diagram*

No.	Simbol	Deskripsi
1.		Kelas pada struktur sistem.
2.	Antar Muka/Interface  Nama_Interface	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3.	Asosiasi / Asociation 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan simbol

4.	Asosiasi Berarah / Directed Association 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan symbol.
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6.	Agregasi / aggregation 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

Sumber: (Rosa and Shalahuddin, 2019)

2.7 Pengujian ISO 25010

ISO/IEC 25010 merupakan model kualitas sistem dan perangkat lunak yang menggantikan ISO/IEC 9126 tentang *software engineering* (International Organisation for Standardisation, 2011). Product quality ini juga digunakan untuk tiga model kualitas yang berbeda untuk produk perangkat lunak antara lain:

- a. Kualitas dalam model penggunaan.
- b. Model kualitas produk.
- c. Data model kualitas.

Model kualitas produk terdiri dari delapan karakteristik yang berhubungan dengan sifat statis perangkat lunak dan sifat dinamis dari sistem komputer. Model ini berlaku untuk sistem komputer dan produk perangkat lunak. Karakteristik yang didefinisikan oleh kedua model tersebut relevan untuk semua produk perangkat lunak dan sistem komputer. Karakteristik dan subkarakteristik memberikan terminologi yang konsisten untuk menentukan, mengukur dan mengevaluasi kualitas sistem dan perangkat lunak. Mereka juga menyediakan seperangkat karakteristik kualitas yang sesuai dengan persyaratan kualitas yang dapat dibandingkan untuk kelengkapan.

2.7.1 *Functional Suitability*

Sejauh mana perangkat lunak mampu menyediakan fungsi yang memenuhi kebutuhan yang dapat digunakan dalam kondisi tertentu. Karakteristik ini dibagi menjadi beberapa karakteristik yaitu.

- a. *Functional completeness*, sejauh mana fungsi yang disediakan mencakup semua tugas dan tujuan pengguna secara spesifik.
- b. *Functional correctness*, sejauh mana produk atau sistem menyediakan hasil yang benar sesuai kebutuhan.
- c. *Functional appropriateness*, sejauh mana fungsi yang disediakan mampu memfasilitasi penyelesaian tugas dan tujuan tertentu.

2.7.2 Compatibility

Sejauh mana sebuah produk, sistem atau komponen dapat bertukar informasi dengan produk, sistem atau komponen dan/atau menjalankan fungsi lain yang diperlukan secara bersamaan ketika berbagi perangkat keras dan environment perangkat lunak yang sama. Karakteristik ini dibagi menjadi 2 karakteristik yaitu.

- a. *Co-existence*, sejauh mana produk atau sistem dapat menjalankan fungsi yang dibutuhkan secara efisien sementara berbagi sumber daya dengan produk atau sistem yang lain tanpa merugikan produk atau sistem tersebut.
- b. *Interoperability*, sejauh mana dua atau lebih produk, sistem atau komponen dapat bertukar informasi dan menggunakan informasi tersebut.

2.7.3 Usability

Sejauh mana sebuah produk atau sistem dapat digunakan oleh user tertentu untuk mencapai tujuan dengan efektif, efisiensi, dan kepuasan tertentu dalam konteks penggunaan. Karakteristik ini terbagi menjadi beberapa karakteristik yaitu.

- a. *Appropriateness recognizability*, sejauh mana pengguna dapat mengetahui apakah sistem atau produk sesuai kebutuhan mereka.
- b. *Learnability*, sejauh mana produk atau sistem dapat digunakan oleh pengguna untuk mencapai tujuan tertentu yang belajar menggunakan sistem atau produk dengan efisien, efektif, kebebasan dari resiko dan kepuasan dalam konteks tertentu.
- c. *Operability*, sejauh mana produk atau sistem mudah dioperasikan dan dikontrol.
- d. *User error protection*, sejauh mana produk atau sistem melindungi pengguna terhadap membuat kesalahan.

- e. *User interface aesthetics*, sejauh mana antarmuka pengguna dari produk atau sistem memungkinkan interaksi yang menyenangkan dan memuaskan pengguna.
- f. *Accessibility*, sejauh mana produk atau sistem dapat digunakan oleh semua kalangan untuk mencapai tujuan tertentu sesuai konteks penggunaan.

2.7.4 Reliability

Sejauh mana sebuah sistem, produk atau komponen dapat menjalankan fungsi tertentu dalam kondisi tertentu selama jangka waktu yang ditentukan. Karakteristik ini terbagi menjadi beberapa subkarakteristik yaitu.

- a. *Maturity*, sejauh mana produk atau sistem mampu memenuhi kebutuhan secara handal di bawah keadaan normal.
- b. *Availability*, sejauh mana produk atau sistem siap beroperasi dan dapat diakses saat perlu digunakan.
- c. *Fault tolerance*, sejauh mana produk atau sistem tetap berjalan sebagaimana yang dimaksud meskipun terjadi kesalahan pada perangkat keras atau perangkat lunak.
- d. *Recoverability*, sejauh mana produk atau sistem mampu dapat memulihkan data yang terkena dampak secara langsung dan menata ulang kondisi system seperti yang diinginkan ketika terjadi gangguan.

2.7.5 Security

Sejauh mana sebuah produk atau sistem melindungi informasi dan data sehingga seseorang atau sistem lain dapat mengakses data sesuai dengan jenis dan level otorisasi yang dimiliki. Karakteristik ini terbagi menjadi beberapa karakteristik yaitu.

- a. *Confidentiality*, sejauh mana produk atau perangkat lunak memastikan data hanya bisa diakses oleh mereka yang berwenang untuk memiliki akses.
- b. *Integrity*, sejauh mana produk atau perangkat lunak mampu mencegah akses yang tidak sah untuk memodifikasi data.
- c. *Non-repudiation*, sejauh mana peristiwa atau tindakan dapat dibuktikan telah terjadi, sehingga tidak ada penolakan terhadap peristiwa atau tindakan tersebut.

- d. *Accountability*, sejauh mana tindakan dari suatu entitas dapat ditelusuri secara unik untuk entitas.
- e. *Authenticity*, sejauh mana identitas subjek atau sumber daya dapat terbukti menjadi salah satu yang diklaim.

2.7.6 Portability

Sejauh mana keefektifan dan efisiensi sebuah sistem, produk atau komponendapat dipindahkan dari satu perangkat keras, perangkat lunak atau digunakan pada lingkungan yang berbeda. Karakteristik ini dibagi menjadi beberapa karakteristik yaitu.

- a. *Adaptability*, sejauh mana produk atau sistem dapat secara efektif dan efisien disesuaikan pada perangkat lunak, perangkat keras dan lingkungan yang berbeda.
- b. *Installability*, sejauh mana produk atau sistem dapat berhasil dipasang atau dihapus dalam lingkungan tertentu.
- c. *Replaceability*, sejauh mana produk atau sistem dapat menggantikan produk atau sistem lain yang ditentukan untuk tujuan yang sama pada lingkungan yang sama.

2.7.7 Performance Efficiency

Kinerja relatif terhadap sumber daya yang digunakan dalam kondisi tertentu. Karakteristik ini terbagi menjadi beberapa subkarakteristik yaitu.

- a. *Time behaviour*, sejauh mana respon dan pengolahan waktu produk atau sistem dapat memenuhi persyaratan ketika menjalankan fungsi.
- b. *Resource utilization*, sejauh mana jumlah dan jenis sumber daya yang digunakan oleh produk atau sistem dapat memenuhi persyaratan ketika menjalankan fungsi.
- c. *Capacity*, sejauh mana batas maksimum parameter produk atau sistem dapat memenuhi persyaratan.

2.7.8 Maintainability

Sejauh mana keefektifan dan efisiensi dari sebuah produk atau sistem dapat dirawat. Karakteristik ini terbagi menjadi beberapa subkarakteristik yaitu.

- a. *Modularity*, sejauh mana sistem terdiri dari komponen terpisah sehingga perubahan atau modifikasi pada salah satu komponen tersebut memiliki dampak yang kecil terhadap komponen yang lain.
 - b. *Reusability*, sejauh mana aset dapat digunakan lebih oleh satu sistem atau digunakan untuk membangun aset lain.
 - c. *Analyzability*, tingkat efektivitas dan efisiensi untuk mengkaji dampak perubahan pada satu atau lebih bagian-bagian produk atau sistem, untuk mendiagnosis kekurangan atau penyebab kegagalan produk, untuk mengidentifikasi bagian yang akan diubah.
 - d. *Modifiability*, sejauh mana produk atau sistem dapat dimodifikasi secara efektif dan efisien tanpa menurunkan kualitas produk yang ada.
- Testability*, tingkat efektivitas dan efisiensi untuk membentuk kriteria uji dari produk, sistem atau komponen dan uji dapat dilakukan untuk menentukan apakah kriteria tersebut telah terpenuhi.