BAB II
TINJAUAN PUSTAKA

2.1 Sistem Informasi

Sistem adalah sebuah hubungan kerja dari berbagai kebijakan yang saling terikat dan berasosiasi untuk mencapai sebuah tujuan yang diinginkan. Sistem informasi adalah sistem dapat didefinisikan dengan mengumpulkan, memproses, menyimpan, menganalisis, menyebarkan informasi untuk tujuan tertentu. Sistem informasi adalah sekumpulan subsistem yang saling berhubungan (Yuliawati, Andriyadi and Nursiyanto, 2022). Sistem juga diartikan untuk berkumpul bersama-sama dan membentuk satu kesatuan, saling berintegrasi dan bekerjasama antara bagian satu dengan yang lainnya dengan cara-cara tertentu untuk melakukan fungsi pengolahan data, menerima masukan (input) berupa data-data, kemudian mengolahnya(processing), dan menghasilkan keluaran (output) berupa informasi sebagai dasar pengambilan (Hakim dan Anshori, 2019).

2.2 Informasi

Informasi merupakan data yang diadaptasi menjadi struktur yang lebih bermanfaat dan lebih berguna bagi yang mendapatkannya, sedangkan data merupakan suatu asal informasi yang mendefinisikan suatu peristiwa yang nyata (Kurniawan and Bodowoso, 2019). Informasi adalah kumpulan data yang sudah diadaptasikan sesuai dengan keputusan tertentu agar dapat menimbulkan suatu hal yang lebih berarti bagi penggunanya dan dapat difungsikan sebagai alasan menentukan ketetapan (Bachry and Yuliawati, 2018).

2.3 Akademik

Akademik adalah sebuah aktivitas yang menganugrahkan penyajian yang berupa data dan mempunyai tingkatan didalam melakukan pemrosesannya untuk menghasilkan informasi yang memiliki arti didalam dunia pendidikan (Agarina and Karim, 2019). Akademi adalah suatu institusi pendidikan tinggi, penelitian, atau keanggotaan kehormatan dan kemampuan yang diperoleh dari hal-hal yang bersifat di luar ilmiah dan jauh dari teori-teori (Maman and Apdian, 2019).

2.4 Sistem Informasi Akademik

Sistem informasi akademik adalah sebuah sistem yang memberikan pelayanan informasi yang berupa data, yang ada kaitannya dengan informasi dunia akademik. Sistem informasi akademik dapat digunakan untuk mengelola maupun mempermudah dalam menjalankan hal yang berkaitan dengan administrasi akademik berupa penerimaan siswa baru, penjadwalan, pengelolaan tenaga pendidik, siswa hingga nilai dengan pemanfaatan teknologi informasi (Pangaribuan and Subakti, 2019).

2.5 *Website*

Website merupakan halaman yang menampilkan informasi data teks, gambar, suara, video atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis. Halaman pertama sebuah website disebut homepage (Apriyanto and Putra, 2020). *Website*/Situs merupakan kumpulan informasi atau kumpulan halaman/page yang bisa diakses lewat jalur internet. Setiap orang di berbagai tempat dan segala waktu bisa menggunakannya selama terhubung secara *online* (Oetomo and Maharginono, 2020).

2.6 PHP

PHP (*Personal Home Page*) adalah pemrograman (interpreter) yang melakukan proses penerjemahan baris sumber menjadi kode mesin yang dimengerti oleh komputer secara dinamis (Muhammad Hakiki, 2021). Pengertian PHP juga merupakan singkatan dari *Hypertex Preprocessor* dengan Bahasa yang berbentuk skrip yang bersifat server side yang dimana proses pengerjaan kode program dilalukan di server, dan hasilnya akan ditampilkan di browser. Sehingga PHP merupakan bahsa pemprograman yang digunakan oleh pengembang untuk membuat sistem *website* dengan kumpulan bahasa HTML dan *script* lainya (Oetomo and Maharginono, 2020).

2.7 MySql

MySQL adalah sebuah database management system (manajemen basis data) menggunakan perintah dasar SQL (*Structured Query Language*) yang cukup terkenal. Database management system (DBMS) MySQL multi pengguna dan

bersifat gratis. Mysql digunakan sebagai wadah dalam mengelola data yang dapat disimpan didigunakan kembali dengan cara yang lebih efesien. *MySQL* adalah singkatan dari *Structue Query Language* yang digunakan untuk mendefinisikan structure data, memodifikasi data pada basis data, menspesifikasi batasan keamanan (*security*), hingga pemeliharaan data (Oetomo and Maharginono, 2020).

2.8 *CodeIgniter*

CodeIgniter adalah *Framework* untuk bahasa pemprograman 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 mengembangkan dalam perangkat *web*, dekstop maupun *mobile* (Raharjo, 2018).

2.9 Metode Pengembang Sistem

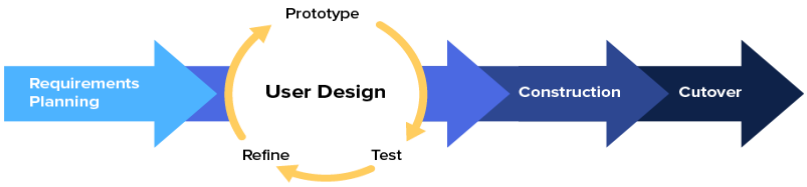
Metode pengembang sistem merupakan metode yang digunakan sebagai alur proses dalam pengembangan, sehingga penelitian dapat di kembangkan sesuai tahapan dari metode pengembang sistem.

2.3.1. *Rapid Application Development* (RAD)

Metode yang berfokus pada pengembangan aplikasi secara cepat, melalui pengulangan dan *feedback* berulang-ulang. RAD diajukan oleh IBM pada tahun 1980 sampai 1990-an, ketika permintaan terhadap aplikasi semakin meningkat (Pratama and Febryansyah, 2021). Dengan banyaknya *demand*, orang-orang di dunia IT harus mencari solusi untuk memenuhi permintaan tersebut. Metode ini merupakan semacam cikal bakal *agile project management*, karena bisa mengikuti *pace* bisnis yang terus berkembang dan juga kebutuhan pasar yang terus meningkat. Pengembangan software pada umumnya seperti *waterfall* model membutuhkan perencanaan yang terbilang cukup kaku. Klien atau pelanggan seakan ‘dipaksa’ untuk menyetujui banyak hal di awal, tetapi mereka tidak bisa melihat proses pembuatannya (Rosa and Shalahuddin, 2019).

Keuntungan utama menjalankan *rapid application development* adalah jangka waktu pengembangan lebih cepat. Hal ini dikarenakan *feedback* dari pelanggan

cepat didapatkan dan semua perubahan yang dilakukan akan sesuai hasil tersebut. Akan tetapi, salah satu kekurangan RAD adalah kamu membutuhkan tim berisikan *developer* yang benar-benar memiliki *skill* tinggi dan juga metode ini hanya bisa digunakan untuk proyek yang bisa termodulasi.



Gambar 2.1 *Rapid Application Development* (RAD)
Sumber: (Rosa and Shalahuddin, 2019)

1. Kelebihan Model RAD

   Kelebihan metodologi RAD menurut Marakas (2006):

   a. Penghematan waktu dalam keseluruhan fase projek dapat dicapai.
   b. RAD mengurangi seluruh kebutuhan yang berkaitan dengan biaya projek dan sumberdaya manusia.
   c. RAD sangat membantu pengembangan aplikasi yang berfokus pada waktu penyelesaian projek.
   d. Perubahan desain sistem dapat lebih berpengaruh dengan cepat dibandingkan dengan pendekatan SDLC tradisional.
   e. Sudut pandang user disajikan dalam sistem akhir baik melalui fungsi-fungsi sistem atau antarmuka pengguna.
   f. RAD menciptakan rasa kepemilikan yang kuat di antara seluruh pemangku kebijakan projek.

2. Kelemahan Model RAD

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

   a. Dengan metode RAD, penganalisis berusaha mepercepat projek dengan terburu-buru.
   b. Kelemahan yang berkaitan dengan waktu dan perhatian terhadap detail. Aplikasi dapat diselesaikan secara lebih cepat, tetapi tidak mampu

mengarahkan penekanan terhadap permasalahan-permasalahan perusahaan yang seharusnya diarahkan.

c. RAD menyulitkan *programmer* yang tidak berpengalaman menggunakan prangkat ini di mana *programmer* dan *analyst* dituntut untuk menguasai kemampuan-kemampuan baru sementara pada saat yang sama mereka harus bekerja mengembangkan sistem

2.3.2 Tahapan Penelitian

Tahapan dalam penelitian sebagai langkah-langkah penelitian yang harus dikerjakan, berikut adalah tahapan penelitian Rapid Application Development.

1. Tahap *Requirements Project*

   RAD dimulai dengan menentukan kebutuhan sebuah proyek (*project requirements*). Pada tahap ini, tim perlu menentukan kebutuhan yang ingin dipenuhi dari sebuah proyek. Kebutuhan ini tidak perlu spesifik. Tapi, sifatnya benar-benar umum dan jumlahnya bisa banyak. Baru dari situ, tim akan menentukan mana kebutuhan yang perlu diprioritaskan. Setelah mendapatkan kebutuhan yang jelas, barulah tim menentukan hal-hal yang lebih detail. Misalkan seperti tujuan, timeline, dan budget yang diperlukan.

2. Tahap Membuat *Prototype*

   Hal yang selanjutnya dilakukan adalah membuat prototype. Developer secepat mungkin akan membuat prototype dari aplikasi yang diinginkan. Lengkap dengan fitur dan fungsi yang berbeda-beda. Tujuannya, sekadar untuk mengecek apakah *prototype* yang dibuat sudah sesuai dengan kebutuhan klien. Meski begitu, tahap ini bisa saja dilakukan berulang-ulang. Kadang juga melibatkan user untuk testing dan memberikan feedback. Proses ini memungkinkan tim mempelajari error yang mungkin muncul ke depannya. Ini berguna untuk mengurangi error dan debugging. Lewat tahapan ini, tim developer memiliki modal untuk membuat aplikasi yang mudah dipakai, stabil, tidak sering error, dan desainnya pun oke.

3. Proses Pengembangan dan Penngumpulan *Feedback*

   Setelah tahu aplikasi seperti apa yang ingin dibuat, developer mengubah *prototype* ke bentuk aplikasi versi beta sampai dengan final. Jadi, bisa dibilang tahap RAD inilah yang cukup intens. Developer terus-menerus

melakukan coding aplikasi, melakukan testing sistem, dan integrasi dengan bagian-bagian lainnya. Karena itulah, developer menggunakan tools dan framework yang mendukung RAD agar cepat. Apalagi proses ini terus diulang sambil terus mempertimbangkan feedback dari klien. Baik itu soal fitur, fungsi, interface, sampai keseluruhan aspek dari produk yang dibuat. Nah, kalau prosesnya berjalan lancar, developer akan melanjutkan ke langkah berikutnya. Yaitu, finalisasi produk atau implementasi. Kalau pun tidak, proses ini kemungkinan akan terus diulang. Pun, kalau apes-apesnya aplikasi tidak tidak menjawab kebutuhan, developer akan kembali ke proses *prototyping*.

4. Tahap *Implementation* (implementasi).

   merupakan tahap pengujian terhadap aplikasi yang dikembangkan. Tahap ini programmer mengembangkan desain menjadi suatu program kemudian dilakukan proses pengujian untuk memeriksa kesalahan sebelum diaplikasikan.

2.10 *Unified Modelling Language* (UML)

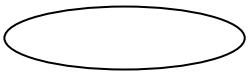
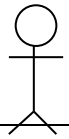
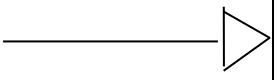
Alat pengembang sistem merupakan konsep desain yang digunakan untuk menggambarkan sistem dengan menggunakan diagram. Penyesuaian alat yang digunakan harus sesuai dengan metode pengembangan yang dilakukan salah satunya adalah penerapan *Unified Modelling Language.* Menurut (Rosa and Shalahuddin, 2019), *Unified Modelling Language* adalah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung (Sutedi, Tinggi and Garut, 2017). Berikut ini merupakan penjelasan tentang masing-masing diagram yang ada pada *Unified Modelling Language.*

2.10.1 *Use Case Diagram*

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 (Rosa dan Salahuddin, 2019). Berikut

simbol-simbol yang akan digunakan dalam menggambarkan *Use Case Diagram* dapat dilihat pada Tabel 2.1.

Tabel 2.1 Simbol *Use Case Diagram*

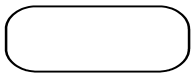
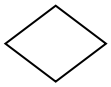
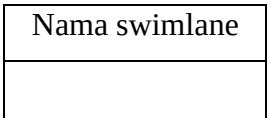
No	Simbol	Deskripsi
1.		*Use case*: Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal *frase* nama *use case*.
2.		Aktor: seseorang/sesuatu yang berinteraksi dengan yang akan dibuat. diluar 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 *(generalization):* merupakan hubungan (umum – khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum
5.	<< Include >>	Include berarti *use case* yang ditambahkan akan dipanggil saat use case tambahan dijalankan.
6.	<<Extend>>	Ekstensi *(extend)* merupakan use case tambahan ke sebuah *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu.

2.10.2 *Activity Diagram*

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 (Rosa dan Salahuddin, 2019). Berikut simbol-simbol yang akan digunakan dalam menggambarkan *activity diagram* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Simbol *Activity Diagram*

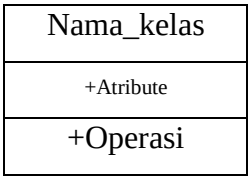
No.	Simbol	Keterangan
1.		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.

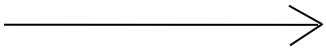
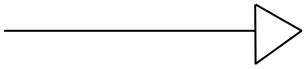
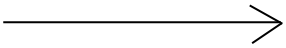
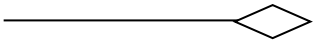
No.	Simbol	Keterangan
2.		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.		Percabangan *(Decision)* merupakan asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.		Penggabungan *(Join)* merupakan asosiasi penggabungan dimana lebih dari satu aktivitas 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.

2.10.3 *Class Diagram*

Class diagram mengembangkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem (Rosa dan Salahuddin, 2019). Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Class Diagram* dapat dilihat pada Tabel 2.3.

Tabel 2.3 Simbol *Class Diagram*

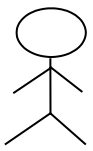
No.	Simbol	Deskripsi
1.		Kelas pada struktur sistem**.**
2.		Sama dengan konsep interface dalam pemrograman berorientasi objek.

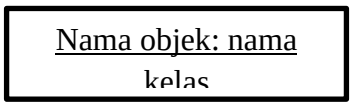
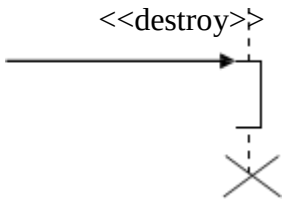
No.	Simbol	Deskripsi
3.	Asosiasi / Asociation	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *symbol*
4.	Asosiasi Berarah / *Digunakan Association* 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *symbol*.
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6.	Ketergantungan / dependency 	Relasi antar kelas dengan makna ketergantungan antar kelas.
7.	Agregasi / *aggregation* 	Relasi antar kelas dengan makna semua bagian *(whole-part)*

2.10.4 *Squence* Diagram

Diagram *squence* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan *massage* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Tabel 2.4 Simbol-simbol *Squence* Diagram

No.	Simbol	Deskripsi
1	Aktor  Atau Nama aktor Tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari akor adalah gambar orang, tapi akor belum tentu merupakan orang, biasanya dinyatakan dalam menggunakan

No.	Simbol	Deskripsi
		kata benda di awal frase nama aktor
2.	Garis hidup /*lifeline* 	Menyataan kehidupan suatu objek
3.	Objek 	Menyetakan objek yang berinteraksi peran
4	Pesan tipe *destroy* 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada *create* maka ada D*estroy*

Sumber: (Rosa dan Salahuddin, 2019)

2.11 Metode Pengujian Sistem

Metode pengujian sistem merupakan metode yang digunakan untuk melakukan testing pada sistem yang dibangun sehingga di peroleh hasil berupa sistem yang sesuai fungsinya.

2.11.1 *Black Box*

Black Box Testing merupakan pengujian yang dapat dilakukan dengan melakukan pengamatan, pada hasil eksekusi melalui beberapa data uji dan memeriksa fungsional yang terdapat pada perangkat lunak. Jadi dapat kita dianalogikan seperti halnya kita melihat ke dalam kotak hitam, sehingga kita hanya bisa melihat tampilan luarnya saja tanpa kita tau apa yang ada didalam kotak hitam tersebut.

Sehingga sama seperti halnya dengan *Black Box Testin*g yang hanya dapat mengevaluasi dari tampilan luarnya dan fungsionalitasnya. Tanpa harus mengetahui apa sesungguhnya yang terjadi dalam proses detilnya. Pada pengetahuan khusus dari struktur kode internal dan pengetahuan pada pemrograman dasar pada umumnya tidak diperlukan untuk *Black Box Testing*. Uji

pada kasus yang dibangun disekitar spesifikasi dan persyaratan, yakni pada aplikasi yang seharusnya dilakukan.

2.11.2 Tahapan Pengujian Sistem

Tahapan pengujian sistem digunakan untuk mengetahui proses pengujian yang akan dilakukan, berikut adalah tahapan pengujian *Black Box*:

1. *Descision Table*

   Decision Table merupakan cara yang tepat untuk memodelkan logika yang cukup rumit, seperti diagram alur, *if-then-else* dan *switch* laporan kasus. Dalam kondisi ini mengaitkan dengan tindakan untuk melakukan, akan tetapi banyak kasus melakukannya dengan cara yang lebih elegan.

2. *All-Pairs Testing*

   All-Pairs Testing atau disebut pairwise testing merupakan metode pengujian perangkat lunak kombinatorial yang digunakan untuk setiap pasangan parameter yang masuk kedalam sistem atau algoritma yang ada pada perangkat lunak.

3. *State Transition Table*

   State Transition Table merupakan teori automata dan logika skuensial, pada table yang menunjukan *state* dalam pengujian. Pada dasarnya sebuah table *state* merupakan table kebenaran yang digunakan untuk beberapa *input* dan *output* termasuk dengan *state* berikutnya dengan kondisi yang sebenarnya terjadi.

4. *Equivalence Partitioning*

   Equicalence Partitioning merupakan teknik yang membagi data masukan dari beberapa unit perangkat lunak menjadi beberapa partisi data dari mana *test case* dapat diturunkan. Pada prinsipnya, uji kasus ini dirancang untuk menutupi setiap partisi minimal.

   Teknik ini digunakan untuk mendefinisikan kasus pengujian yang mengungkap kelas kesalahan, sehingga mengurangi jumlah pengujian yang harus dilakukan.

5. *Boundry Values Analysis*

   Boundary Value Analysis merupakan Pengujian yang dirancang untuk mencakup perwakilan dari batas Nilai-nilai batas. Pada nilai-nilai di sebuah partisi kesetaraan atau sebesar nilai terkecil di kedua sisi tepi.

2.12 Penelitian Terdahulu

Penelitian terdahulu bertujuan untuk mendapatkan bahan perbandingan dan acuan. Selain itu, untuk menghindari anggapan kesamaan dengan penelitian ini. Maka dalam kajian pustaka ini peneliti mencantumkan hasil-hasil penelitian terdahulu sebagai berikut:

Tabel 2.5 Penelitian Terdahulu

Nama	Judul	Variabel	Metode Analisis	Hasl Analisis
(Pangaribuan and Subakti, 2019)	Sistem Informasi Akademik Berbasis Web pada SMK (Sekolah Menengah Kejuruan) Teknologi Industri Pembangunan Cimahi	Sistem Informasi Akademik	Prototype	Dapat mempermudah pihak tata usaha pada saat melakukan pengecekan persyaratan pendaftaran siswa baru, dan dapat mempersingkat waktu rekapitulasi data calon siswa, selain itu bagian kurikulum dapat dengan mudah melakukan pembuatan penjadwalan mata pelajaran setiap kelas tanpa adanya bentrokan, kemudian untuk seluruh guru dapat dengan mudah menginputkan data penilaian siswa yang terintegrasi dengan basis data sehingga setiap wali kelas bisa mencetak rapor tanpa harus menyalin data nilai yang diberikan oleh setiap g
(Megawaty *et al.*, 2020)	Sistem Monitoring Kegiatan Akademik Siswa Menggunakan Website	Sistem Monitoring Kegiatan Akademik	Prototype	Adanya sistem monitoring akademik siswa dapat membantu guru dalam melaporkan kegiatan akademik siswa sehingga membantu orang tua atau wali murid dalam proses monitoring kegiatan siswa

Nama	Judul	Variabel	Metode Analisis	Hasl Analisis
				serta memudahkan pihak sekolah dalam pelaporan kegiatan siswa
(Bachry and Yuliawati, 2018)	Desain Sistem Informasi Akademik di SMU Negeri 1 Pasir Sakti Berbasis Web	Desain Sistem Informasi Akademik	*Waterfall*	Dibangunya sistem informasi ini, memudahkan siswa dalam pendaftaran online dan membantu mempermudah pengolahan data akademik sekolah lebih maksimal dan praktis karena dapat diakses dari mana saja sekaligus menjaga data tetap aman.Selain itu bisa dapat memudahkan pihak sekolah untuk mengelola sistem secara professional dalam melayani kebutuhan siswa.
(Nofikasari, Purwanto and ..., 2022)	Penerapan Metode Rapid Application Development (RAD) Dalam Sistem Informasi Anak Putus Sekolah (Siap Sekolah)	Sistem Informasi Anak Putus Sekolah	Rapid Application Development (RAD)	Sistem mengolah serta menghasilkan laporan data anak putus sekolah kategori siswa keluar tidak melanjutkan/ dropout (DO) dan siswa lulus tidak melanjutkan (LTM).