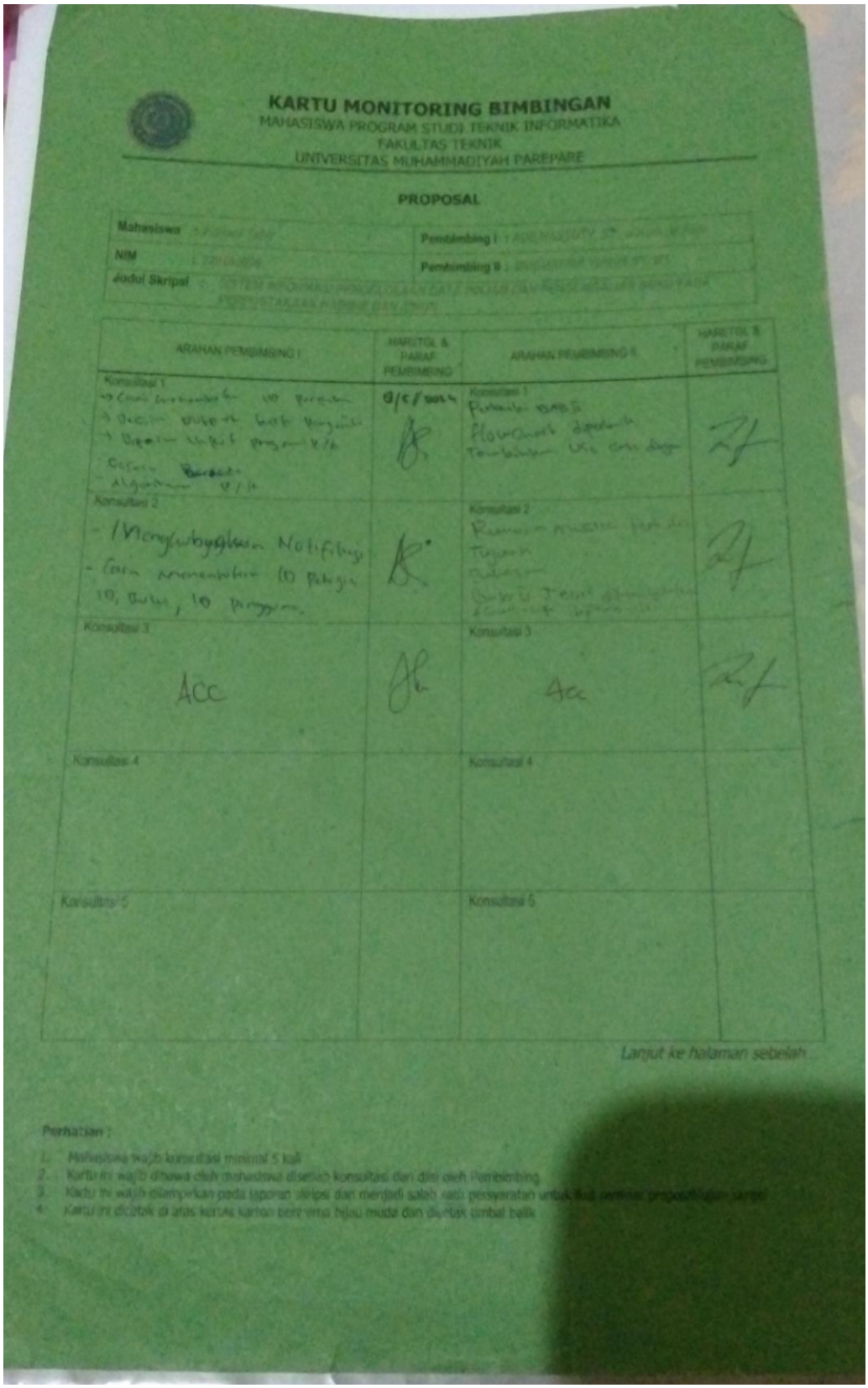
L A M P I R A N

A. Kartu Monitoring Bimbingan Proposal



KARTU MONITORING BIMBINGAN
MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH PAREPARE

PROPOSAL

Mahasiswa	Pembimbing I
NIM	Pembimbing II
Judul Skripsi	

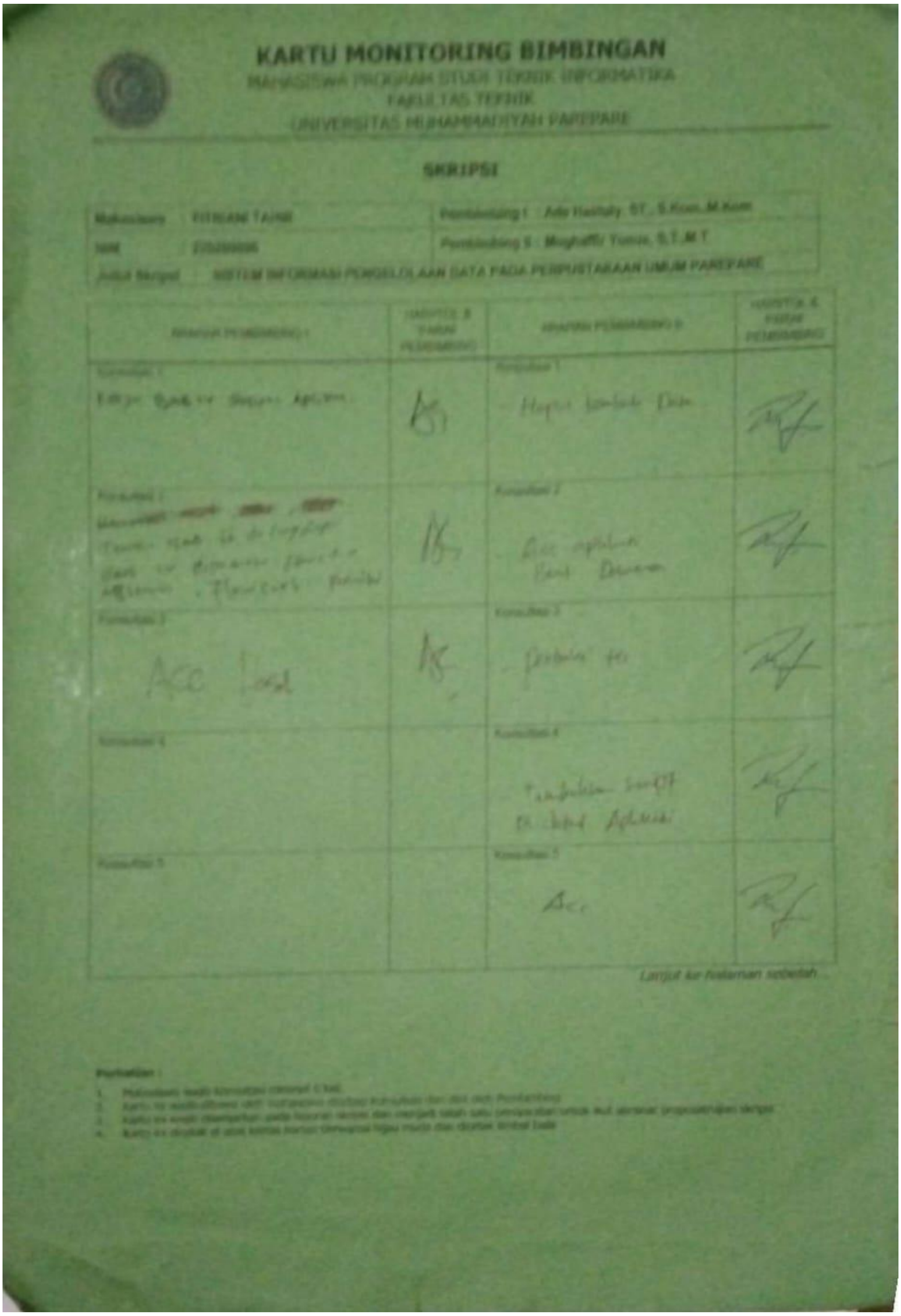
ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1		Konsultasi 1	
Konsultasi 2		Konsultasi 2	
Konsultasi 3		Konsultasi 3	
Konsultasi 4		Konsultasi 4	
Konsultasi 5		Konsultasi 6	

Lanjut ke halaman sebelah

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk
4. Kartu ini dicetak di atas kertas karton berwarna hijau muda dan dicetak bolak balik

B. Kartu Monitoring Bimbingan Skripsi



KARTU MONITORING BIMBINGAN

MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MUHAMMADIYAH PAREPARE

SKRIPSI

Pembimbing I : Ade Hastuty, ST., S.Kom., M.Kom

Pembimbing II : Mughaffir Yunus, S.T., M.T

Judul Skripsi : SISTEM INFORMASI PENGELOLAAN DATA PADA PERPUSTAKAAN UMUM PAREPARE

Script Aplikasi

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;

public class menu : MonoBehaviour
{
   public void pindahScene(string
scenename)
   {

SceneManager.LoadScene(scenename);
   }

   public void BackButton(string
scenename)
   {

SceneManager.LoadScene(scenename);
   }
   public void QuitButton()
   {
      Application.Quit();
      Debug.Log("ANJAY");
   }
}
using System.Collections;
using UnityEngine;
using UnityEngine.UI;

public class PanelClick :
MonoBehaviour
{
   public GameObject targetObject; //
Objek yang ingin di nonaktifkan
   private Button panelButton;     //
Panel yang akan diklik (pastikan
memiliki komponen Button)

   void Start()
   {
      // Mendapatkan komponen Button
using UnityEngine;

public class PopupController :
MonoBehaviour
{
   private Animator animator;
   private bool canShowPopup =
true;

   void Start()
   {
      animator =
GetComponent<Animator>();
   }

   public void ShowPopup()
   {
      if (canShowPopup)
      {

animator.SetTrigger("Show");
         canShowPopup = false;
      }
   }

   public void HidePopup()
   {
      animator.SetTrigger("Hide");
   }

   // Panggil fungsi ini di akhir
animasi "Hide"
   public void EnableShow()
   {
      canShowPopup = true;
   }
}

using System;
using Firebase;
using Firebase.Database;
using Firebase.Extensions;
using TMPro;
using UnityEngine;
```

```
dari panel
        panelButton =
GetComponent<Button>();

        // Menambahkan listener pada
event klik button
        if (panelButton != null)
        {

panelButton.onClick.AddListener(OnP
anelClick);
        }
        else
        {
            Debug.LogError("Button
component not found on this panel.");
        }
    }

    void OnPanelClick()
    {
        // Menonaktifkan objek target saat
panel diklik
        if (targetObject != null)
        {
            targetObject.SetActive(false);
        }
        else
        {
            Debug.LogError("Target object
not assigned.");
        }
    }
}
```

```
using UnityEngine.UI;

public class BookManager :
MonoBehaviour
{
    private DatabaseReference
dbReference;
    public GameObject bookPrefab;
    public Transform contentPanel;

    void Start()
    {
        Debug.Log("Start method
called.");
        InitializeFirebase();
    }

    private void InitializeFirebase()
    {
        Debug.Log("Initializing
Firebase...");
        FirebaseApp

.CheckAndFixDependenciesAsync(
)

.ContinueWithOnMainThread(task
=>
            {
                if (task.Result ==
DependencyStatus.Available)
                {
                    dbReference =
FirebaseDatabase

.GetInstance("https://perpustakaan-
apk-default-rtdb.firebaseio.com/")
                        .RootReference;
                    Debug.Log("Firebase
connected successfully.");
                    GetBookDetails();


Debug.Log($"Processing book ID:
{bookId}");
```

```
                }
                else
                {

Debug.LogError("Firebase connection
failed: " + task.Result);
                }
            });
    }
```

```
    private void GetBookDetails()
    {
        // Membaca koleksi yang dipilih dari PlayerPrefs
        string selectedKoleksi = PlayerPrefs.GetString("SelectedKoleksi", "Fiksi");  // Default ke "Fiksi" jika tidak ada
        Debug.Log($"Fetching books from collection: {selectedKoleksi}");

        // Query Firebase untuk mengambil buku berdasarkan koleksi yang dipilih
        dbReference
            .Child("books")
            .OrderByChild("koleksi")
            .EqualTo(selectedKoleksi) // Filter berdasarkan koleksi
            .GetValueAsync()
            .ContinueWithOnMainThread(task =>
            {
                if (task.IsCompleted)
                {
                    if (task.Exception != null)
                    {
                        Debug.LogError("Exception occurred while fetching books: " + task.Exception);
                        return;
                    }

                    DataSnapshot snapshot = task.Result;

                    if (snapshot.HasChildren)
                    {
                        Debug.Log($"Total books in {selectedKoleksi} collection: {snapshot.ChildrenCount}");

                        foreach (DataSnapshot bookSnapshot in snapshot.Children)
                        {
                            // Instansiasi prefab dan mengisi data buku
                            GameObject newBookItem = Instantiate(bookPrefab, contentPanel);
                            TMP_Text[] texts = newBookItem.GetComponentsInChildren<TMP_Text>();
                            foreach (TMP_Text text in texts)
                            {
                                switch (text.name)
                                {
                                    case "class":
                                        text.text = bookClass;
                                        break;
                                    case "judul":
                                        text.text = judul;
                                        break;
                                    case "penulis":
                                        text.text = penulis;
                                        break;
                                    case "kategori":
                                        text.text = kategori;
                                        break;
                                    case "rak":
                                        text.text = rak;
                                        break;
                                    case "penerbit":
                                        text.text = penerbit;
                                        break;
                                    case "kota_tahun":
                                        text.text =
```

```
                    string bookId =
bookSnapshot.Key;
                    string bookClass =
bookSnapshot.Child("class").Value?.T
oString();
                    string judul =
bookSnapshot.Child("judul").Value?.T
oString();
                    string penulis =
bookSnapshot.Child("penulis").Value?
.ToString();
                    string kategori =
bookSnapshot.Child("kategori").Value
?.ToString();
                    string rak =
bookSnapshot.Child("rak").Value?.To
String();
                    string penerbit =
bookSnapshot.Child("penerbit").Value
?.ToString();
                    string kotaTahun =
bookSnapshot.Child("kota_tahun").Va
lue?.ToString();
                    string halaman =
bookSnapshot.Child("halaman").Valu
e?.ToString();
                    string koleksi =
bookSnapshot.Child("koleksi").Value?
.ToString();
                    int stok =
Convert.ToInt32(bookSnapshot.Child(
"stok").Value ?? 0);
      }
            else
            {
               Debug.LogError("Failed
to retrieve books: " + task.Exception);
            }
         });
   }
}

using System;
using System.Collections;
using System.Collections.Generic;
using Firebase;
kotaTahun;
                        break;
                    case "hal":
                        text.text =
halaman;
                        break;
                    case
"koleksi":
                        text.text =
koleksi;
                        break;
                    case "stok":
                        text.text =
$"{stok}";
                        break;
                    default:
                        break;
                }
            }
        }

        // Pastikan layout
diupdate di main thread

UnityMainThreadDispatcher
            .Instance()
            .Enqueue(() =>
            {

LayoutRebuilder.ForceRebuildLayo
utImmediate(

contentPanel.GetComponent<RectT
ransform>()
                );

Debug.Log("Layout rebuilt after
adding all book items.");
            });
        }
        else
        {
            Debug.Log($"No
books found in the
{selectedKoleksi} collection.");
        }
```

```
using Firebase.Database;
using Firebase.Extensions;
using TMPro;
using UnityEngine;
using Vuforia;

public class ScanBook :
MonoBehaviour
{
    private DatabaseReference
dbReference;
    public GameObject bookPrefab;
    public Transform contentPanel;

    // Tambahkan list
ObserverBehaviour untuk mendukung
banyak target
    public List<ObserverBehaviour>
targetObservers;

    // Flag untuk memastikan Firebase
sudah diinisialisasi
    private bool isFirebaseInitialized =
false;

    // Coroutine reference
    private Dictionary<string,
Coroutine> fetchCoroutines = new
Dictionary<string, Coroutine>();

    // Fetch interval in seconds
    [SerializeField]
    private float fetchInterval = 5f; //
Sesuaikan sesuai kebutuhan

    void Start()
    {
        Debug.Log("[Start] Method
called.");
        InitializeFirebase();
    }

    private void InitializeFirebase()
    {
        Debug.Log("[InitializeFirebase]
Initializing Firebase...");
connection failed:
{dependencyStatus}");
                // Opsional: Beritahu
pengguna melalui UI
            }
        });
    }

    private void InitializeVuforia()
    {
        Debug.Log("[InitializeVuforia]
Initializing Vuforia...");
        if (targetObservers == null ||
targetObservers.Count == 0)
        {

Debug.LogError("[InitializeVuforia
] No targetObservers assigned in the
Inspector.");
            return;
        }

        foreach (var observer in
targetObservers)
        {
            if (observer != null)
            {

observer.OnTargetStatusChanged
+= OnTargetStatusChanged;

Debug.Log($"[InitializeVuforia]
Subscribed to target
'{observer.TargetName}'.");
            }
            else
            {

Debug.LogError("[InitializeVuforia
] One of the targetObservers is
null.");
            }
        }
    }
```

```
FirebaseApp.CheckAndFixDependenciesAsync().ContinueWithOnMainThread(task =>
        {
            var dependencyStatus = task.Result;

Debug.Log($"[InitializeFirebase] Dependency Status: {dependencyStatus}");
            if (dependencyStatus == DependencyStatus.Available)
            {
                FirebaseApp app = FirebaseApp.DefaultInstance;
                dbReference = FirebaseDatabase
                    .GetInstance(app)
                    .GetReference("books"); // Mengarahkan langsung ke node "books"

Debug.Log("[InitializeFirebase] Firebase connected successfully.");
                isFirebaseInitialized = true;
                InitializeVuforia(); // Inisialisasi Vuforia setelah Firebase siap
            }
            else
            {

Debug.LogError($"[InitializeFirebase] Firebase '{behaviour.TargetName}' is not TRACKED.");
            if (fetchCoroutines.ContainsKey(behaviour.TargetName))
            {

Debug.Log($"[OnTargetStatusChanged] Stopping coroutine for target '{behaviour.TargetName}'");
    private void OnTargetStatusChanged(ObserverBehaviour behaviour, TargetStatus status)
    {

Debug.Log($"[OnTargetStatusChanged] Target '{behaviour.TargetName}' status changed to {status.Status}");

        if (status.Status == Status.TRACKED)
        {

Debug.Log($"[OnTargetStatusChanged] Target '{behaviour.TargetName}' is TRACKED.");
            if (isFirebaseInitialized)
            {
                if (!fetchCoroutines.ContainsKey(behaviour.TargetName))
                {

Debug.Log($"[OnTargetStatusChanged] Starting coroutine for target '{behaviour.TargetName}'");

fetchCoroutines[behaviour.TargetName] = StartCoroutine(FetchBookDetailsRepeatedly(behaviour.TargetName));
                }
                else
                {

Debug.Log($"[OnTargetStatusChanged] Coroutine already running for target '{behaviour.TargetName}'");
                }
            }
            else
            {
```

```
StopCoroutine(fetchCoroutines[behaviour.TargetName]);

fetchCoroutines.Remove(behaviour.TargetName);
            }
            // Anda mungkin perlu memodifikasi ClearBookDetails untuk menangani multiple target
            ClearBookDetails();
        }
    }

    private IEnumerator FetchBookDetailsRepeatedly(string targetName)
    {
        Debug.Log($"[Coroutine] Starting FetchBookDetailsRepeatedly for targetName: '{targetName}'");
        while (true)
        {

FetchBookDetailsByTarget(targetName);
            yield return new WaitForSeconds(fetchInterval);
        }
    }

    private void FetchBookDetailsByTarget(string targetName)
    {

Debug.Log($"[FetchBookDetailsByTarget] Called with targetName: '{targetName}'");

        if (string.IsNullOrEmpty(targetName))
        {

Debug.LogError("[FetchBookDetailsByTarget] Target name is null or empty.");
Debug.LogWarning("[OnTargetStatusChanged] Firebase not initialized. Cannot fetch book data.");
            }
        }
        else
        {

Debug.Log($"[OnTargetStatusChanged] Target

bookSnapshot.Child("penulis").Value?.ToString() ?? "N/A";
                string kategori = bookSnapshot.Child("kategori").Value?.ToString() ?? "N/A";
                string rak = bookSnapshot.Child("rak").Value?.ToString() ?? "N/A";
                string penerbit = bookSnapshot.Child("penerbit").Value?.ToString() ?? "N/A";
                string kotaTahun = bookSnapshot.Child("kota_tahun").Value?.ToString() ?? "N/A";
                string halaman = bookSnapshot.Child("halaman").Value?.ToString() ?? "N/A";
                string koleksi = bookSnapshot.Child("koleksi").Value?.ToString() ?? "N/A";
                int stok = 0;
                if (bookSnapshot.Child("stok").Exists)
                {
                    if (int.TryParse(bookSnapshot.Child("stok").Value.ToString(), out int parsedStok))
                    {
                        stok = parsedStok;
                    }
                    else
                    {

Debug.LogWarning($"[FetchBook
```

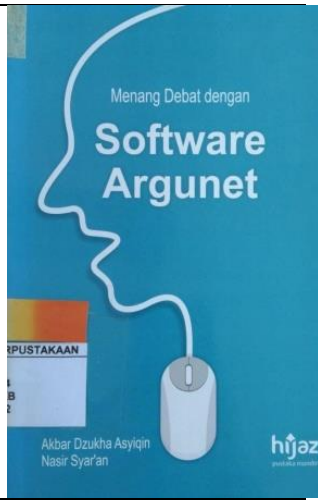
```
            return;
        }


dbReference.Child(targetName).GetV
alueAsync().ContinueWithOnMainThr
ead(task =>
        {
            if (task.IsCompleted)
            {
                if (task.Exception != null)
                {

Debug.LogError($"[FetchBookDetails
ByTarget] Exception:
{task.Exception}");
                    return;
                }

                DataSnapshot bookSnapshot
= task.Result;

                if (bookSnapshot.Exists)
                {

Debug.Log($"[FetchBookDetailsByTa
rget] Found book data for
'{targetName}':
{bookSnapshot.ChildrenCount}
entries");

                    // Ekstrak data buku
dengan pengecekan null
                    string bookClass =
bookSnapshot.Child("class").Value?.T
oString() ?? "N/A";
                    string judul =
bookSnapshot.Child("judul").Value?.T
oString() ?? "N/A";
                    string penulis =
        string penulis,
        string kategori,
        string rak,
        string penerbit,
        string kotaTahun,
        string halaman,
DetailsByTarget] Stok untuk
'{targetName}' tidak dapat diubah
ke integer. Default ke 0.");
                    }
                }


Debug.Log($"[FetchBookDetailsBy
Target] Book Data: {judul},
{penulis}, {kategori}, {rak},
{penerbit}, {kotaTahun},
{halaman}, {koleksi}, {stok}");

                // Tampilkan detail
buku
                DisplayBookDetails(
                    bookClass,
                    judul,
                    penulis,
                    kategori,
                    rak,
                    penerbit,
                    kotaTahun,
                    halaman,
                    koleksi,
                    stok
                );
            }
            else
            {

Debug.LogWarning($"[FetchBook
DetailsByTarget] No book data
found for '{targetName}'. Check
database.");
                ClearBookDetails();
            }
        }
        else
        {

Debug.LogError($"[FetchBookDeta
ilsByTarget] Task not completed.
Exception: {task.Exception}");
        }
    });
```

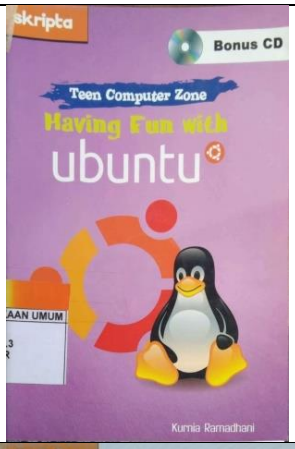
```
        string koleksi,
        int stok
    )
    {

Debug.Log("[DisplayBookDetails]
Displaying book details in UI...");

        if (contentPanel == null)
        {

Debug.LogError("[DisplayBookDetail
s] Content Panel is not assigned.");
            return;
        }

        if (bookPrefab == null)
        {

Debug.LogError("[DisplayBookDetail
s] Book Prefab is not assigned.");
            return;
        }

        // Clear previous entries
        foreach (Transform child in
contentPanel)
        {
            Destroy(child.gameObject);
        }

        // Instantiate a new book item
        GameObject newBookItem =
Instantiate(bookPrefab, contentPanel);
        TMP_Text[] texts =
newBookItem.GetComponentsInChild
ren<TMP_Text>();

        foreach (TMP_Text text in texts)
        {
            switch (text.name.ToLower())
            {
                case "class":
                    text.text = bookClass;
                    break;
                case "judul":
                    text.text = judul;
                    break;
                case "koleksi":
                    text.text = koleksi;
                    break;
                case "stok":
                    text.text =
stok.ToString();
                    break;
                default:

Debug.LogWarning($"[DisplayBoo
kDetails] Unrecognized TMP_Text
name: '{text.name}'");
                    break;
            }
        }


Debug.Log("[DisplayBookDetails]
Book details displayed in UI.");
    }

    private void ClearBookDetails()
    {

Debug.Log("[ClearBookDetails]
Clearing book details from UI...");
        if (contentPanel == null)
        {

Debug.LogError("[ClearBookDetail
s] Content Panel is not assigned.");
            return;
        }

        foreach (Transform child in
contentPanel)
        {
            Destroy(child.gameObject);
        }
    }

    void OnDestroy()
    {
```

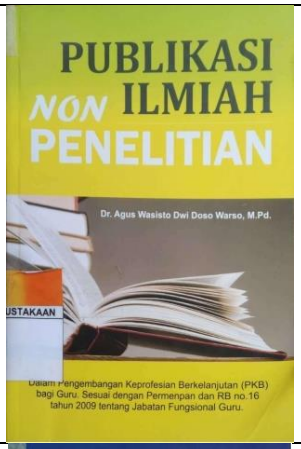
```
                text.text = judul;
                break;
            case "penulis":
                text.text = penulis;
                break;
            case "kategori":
                text.text = kategori;
                break;
            case "rak":
                text.text = rak;
                break;
            case "penerbit":
                text.text = penerbit;
                break;
            case "kota_tahun":
                text.text = kotaTahun;
                break;
            case "hal":
                text.text = halaman;
                break;
            case "koleksi":
                text.text = koleksi;
OnTargetStatusChanged;

Debug.Log($"[OnDestroy]
Unsubscribed from target
'{observer.TargetName}'.");
            }
        }
    }
```

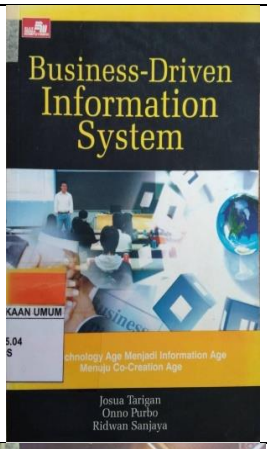
```
    Debug.Log("[OnDestroy]
Called. Unsubscribing from Vuforia
events.");
    if (targetObservers != null)
    {
        foreach (var observer in
targetObservers)
        {
            if (observer != null)
            {

observer.OnTargetStatusChanged -
=

    // Pastikan semua coroutine
dihentikan saat objek dihancurkan
    foreach (var coroutine in
fetchCoroutines.Values)
    {
        if (coroutine != null)
        {
            StopCoroutine(coroutine);
        }
    }
    fetchCoroutines.Clear();
  }
}
```

Data Marker



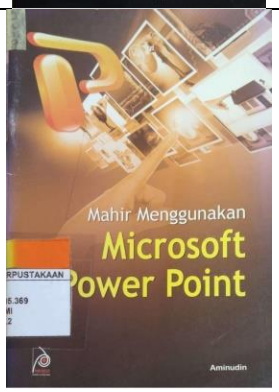

skripta
Bonus CD
Teen Computer Zone
Having Fun with
ubuntu
Kurnia Ramadhani

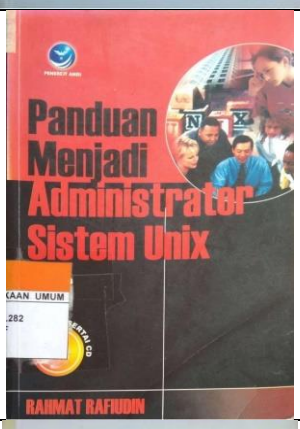

PUBLIKASI
NON ILMIAH
PENELITIAN
Dr. Agus Wasisto Dwi Doso Warso, M.Pd.
Dalam Pengembangan Keprofesian Berkelanjutan (PKB) bagi Guru. Sesuai dengan Permenpan dan RB no.16 tahun 2009 tentang Jabatan Fungsional Guru.

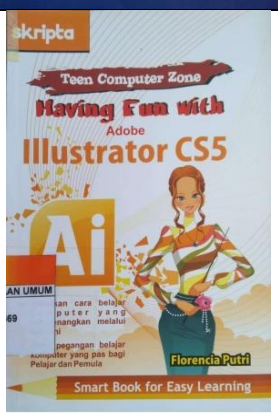

Business-Driven
Information
System
Josua Tarigan
Onno Purbo
Ridwan Sanjaya


skripta
Teen Computer Zone
Having Fun with
Adobe
Flash Professional CS5
Ronald Budi
Smart Book for Easy Learning

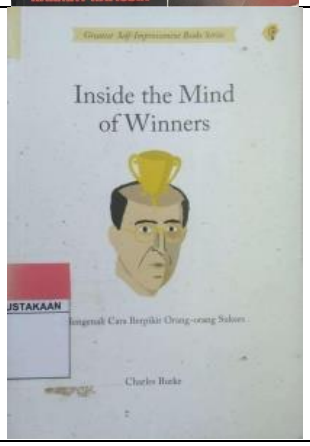

Transaksi
Forex Online
dengan Marketiva
Streamster
Joko Salim, S.Kom, SE

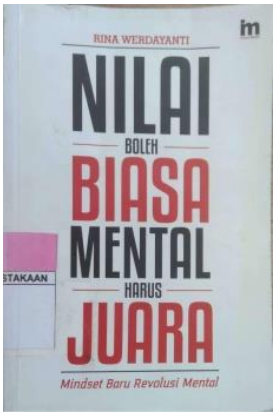

Mahir Menggunakan
Microsoft
Power Point
Aminudin


Panduan
Menjadi
Administrator
Sistem Unix
RAHMAT RAFUDIN

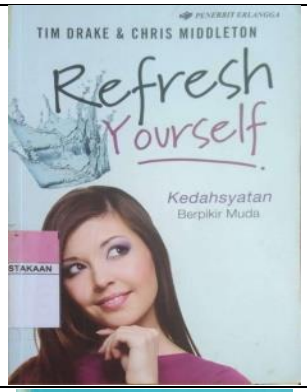

skripta
Teen Computer Zone
Having Fun with
Adobe
Illustrator CS5
Florencia Putri
Smart Book for Easy Learning


bagustakwin
psikologi naratif
Membaca Manusia sebagai Kisah


Inside the Mind
of Winners
Charles Burke


RINA WERDAYANTI
NILAI
BOLEH
BIASA
MENTAL
HARUS
JUARA
Mindset Baru Revolusi Mental


Stiletto
Perkembangan
Bahasa Anak
VIT ARDHYANTAMA
CHUSNA APRIYANTI

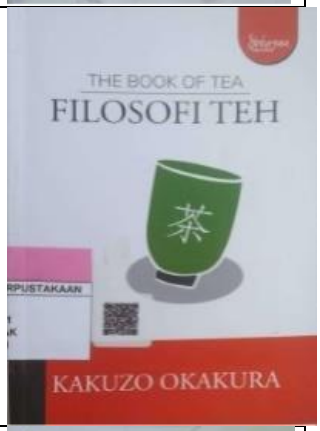

TIM DRAKE & CHRIS MIDDLETON
Refresh Yourself
Kedahsyatan
Berpikir Muda

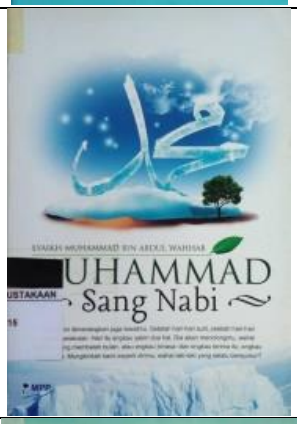

TES
IQ
PUTRA PUTRI ANDA
Mengenal Tes Inteligensi dan Estimasi Prestasi Anak


Menyerah?
Tidak hari ini!!
Nda Ginting

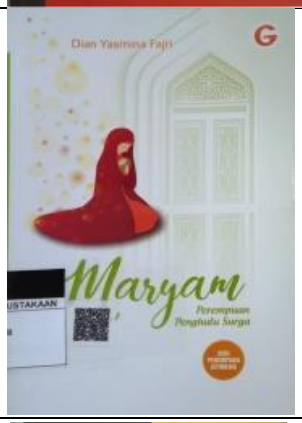

DIARY INTROVERT
Hardy Zhu


DELTA NASUTION
Mengasah
OTAK SETAJAM SILET
Mendongkrak Kemampuan Otak 100 x Lipat
Full Latihan Soal-soal Penalaran Tingkat Tinggi

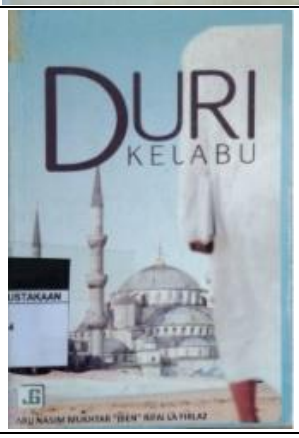

THE BOOK OF TEA
FILOSOFI TEH
KAKUZO OKAKURA

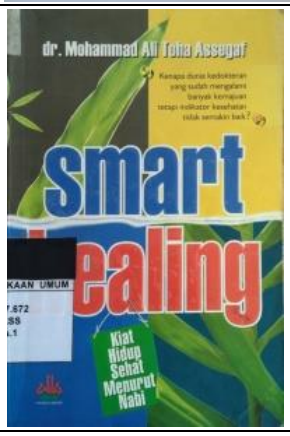

UHAMMAD
Sang Nabi

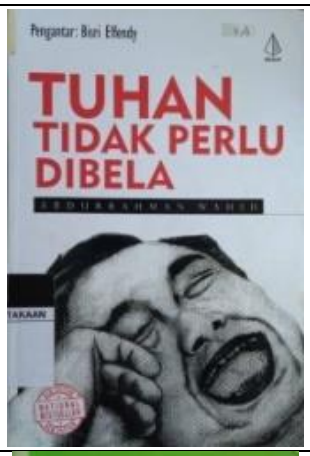

LUKMAN HAKIEM
MERAWAT INDONESIA
Belajar dari Tokoh dan Peristiwa

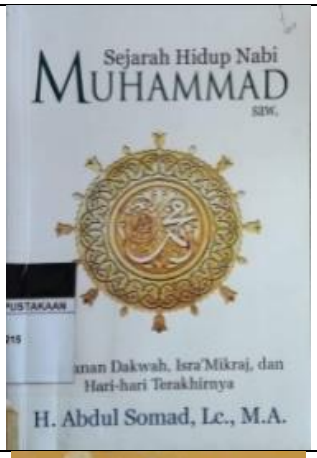

Dian Yasmina Fajri
Maryam


Rofi' Maryam
Berhenti
Sebuah
Alif

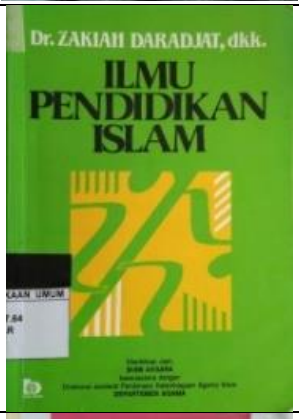

DURI
KELABU


dr. Mohammad Ali Toha Assegaf
smart
ealing
Hidup Sehat Menurut Nabi

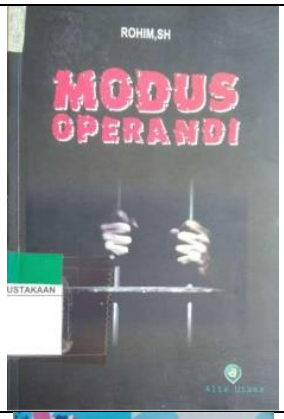

Pengantar: Bari Effendy
TUHAN
TIDAK PERLU
DIBELA

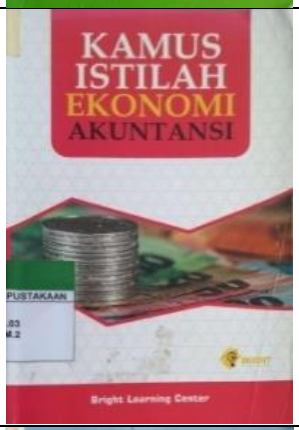

Sejarah Hidup Nabi
MUHAMMAD
saw.
H. Abdul Somad, Lc., M.A.


AULIA FADHLI
Pendidikan
Anak Tuna Laras


Dr. ZAKIAH DARADJAT, dkk.
ILMU
PENDIDIKAN
ISLAM


dari
Kuntum
Menjadi
Bunga
IBNU BASYAR


ROHIM,SH
MODUS
OPERANDI

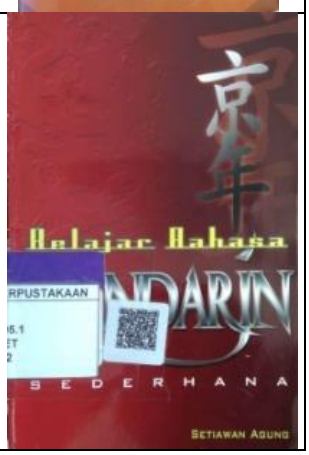

KAMUS
ISTILAH
EKONOMI
AKUNTANSI
Bright Learning Center

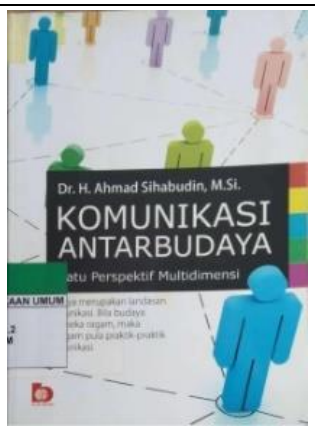

PENGEMBANGAN
LANJUTAN
RUMAH SAKIT
Menuju Rumah Sakit
Kelas Dunia
ROY SUBIROSA SABARGUNA

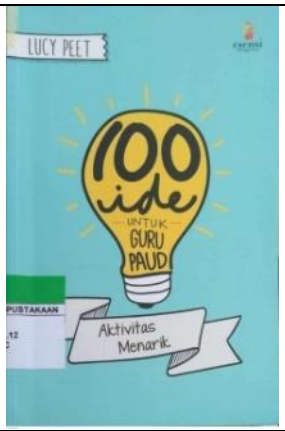

Panca Prastowo
Langkah
Super
Cepat
MENINGKATKAN
KOSAKATA
NGGRIS
Penting bagi Pemula


Pokok-Pokok
PEMERINTAHAN
AERAH
I. WIDARTA

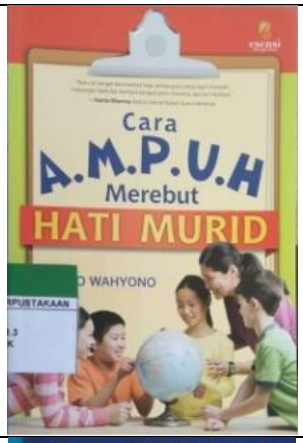

DRADJAD PREMADI, S.T.
POLAMATIKA
TEKNIK BERHITUNG CEPAT & MUDAH
DENGAN MENGGUNAKAN POLA BILANGAN
67 x 8 = ...?
Metode 3D
DIKTE
DENGAN KERAS
DIULANG
67 x 8 = 536
PERKALIAN
PEMBAGIAN
AKAR KUADRAT
PERKALIAN KUADRAT

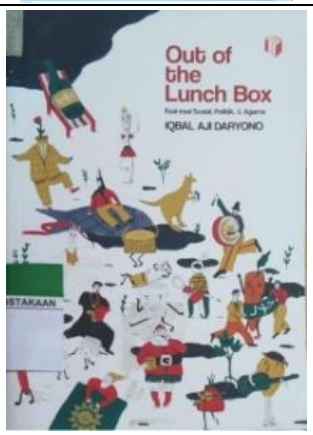

京年
Belajar Bahasa
ANDARIN
SEDERHANA
SETIAWAN AGUNG

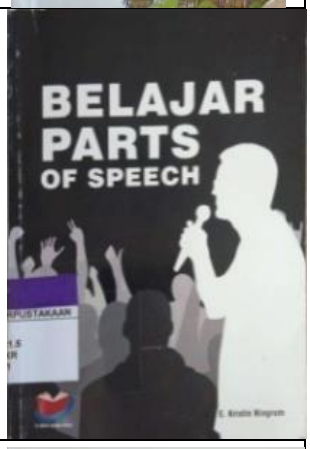

Dr. H. Ahmad Sihabudin, M.Si.
KOMUNIKASI ANTARBUDAYA

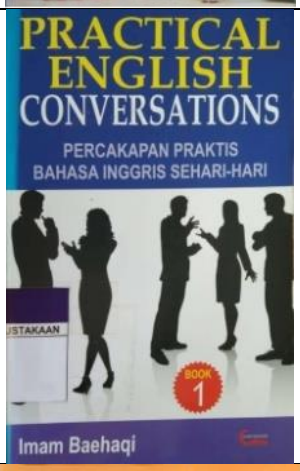

LUCY PEET
100 ide
UNTUK
GURU PAUD
Aktivitas Menarik


LANGSUNG LANCAR
PERCAKAPAN SEHARI-HARI
BAHASA JEPANG

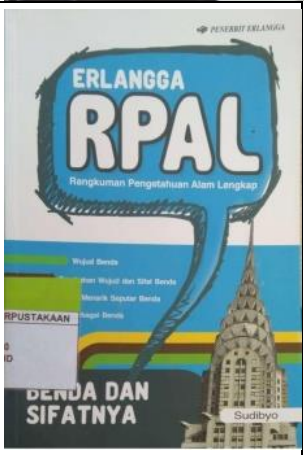

Cara
A.M.P.U.H
Merebut
HATI MURID


Out of the Lunch Box
IQBAL AJI DARYONO

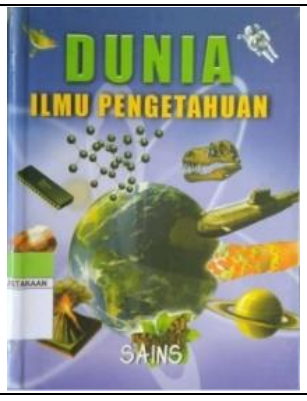

BELAJAR PARTS OF SPEECH


PRACTICAL ENGLISH CONVERSATIONS
PERCAKAPAN PRAKTIS BAHASA INGGRIS SEHARI-HARI
BOOK 1
Imam Baehaqi


BELAJAR GENRE!!
Memahami Teks Procedure, Descriptive, dan Report
E. Kristin Ningrum


ERLANGGA
RPAL
Rangkuman Pengetahuan Alam Lengkap
BENDA DAN SIFATNYA
Sudibyo

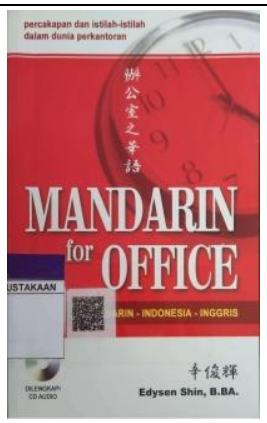

Holben D. Lingga
Buku Seri Persiapan
iBT TOEFL
Tes Kosakata
Mastering iBT TOEFL Exam Vocabulary

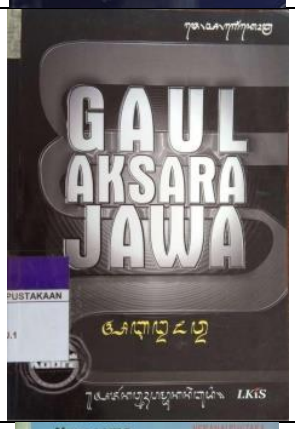

DUNIA
ILMU PENGETAHUAN
SAINS

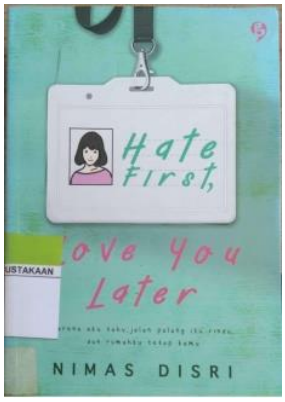

satelit
untuk anak bangsa

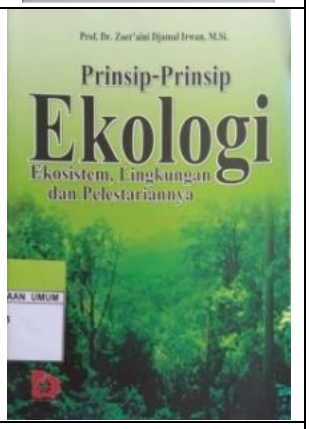

Intisari
Tata Bahasa
Inggris
Kontemporer
nouns
prepositions
participles
conjunctions


Wawan Badrianto
MODUL
CERDIK
MATEMATIKA
SMA/MA
KELAS X, XI, & XII
FREE VIDEO
FREE APPS


percakapan dan istilah-istilah
dalam dunia perkantoran
MANDARIN
for OFFICE
INDONESIA - INGGRIS
Edysen Shin, B.BA.


GAUL
AKSARA
JAWA

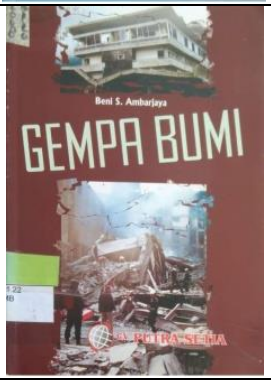

Hate
First,
ove You
Later
NIMAS DISRI

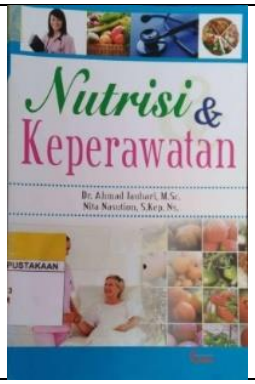

Prof. Dr. Zoer'aini Djamal Irwan, M.Si.
Prinsip-Prinsip
Ekologi
Ekosistem, Lingkungan
dan Pelestariannya


Nanosamo
Cara Mudah
Memahami
Sains Sehari-hari
Penulis: Dr Kebamoto


Buku Ajar
Reni Agustina Harahap, S.ST., M.Kes.
Fauzi Eka Putra, M.I.Kom.


EDISI
2
Buku Saku
MANAJEMEN UNIT
PERSALINAN
LEROY C. EDOZIEN
PENERBIT BUKU KEDOKTERAN
EGC

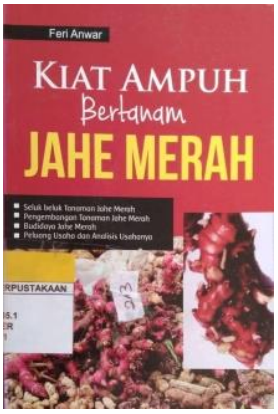

Beni S. Ambarjaya
GEMPA BUMI


Nutrisi &
Keperawatan


Asuhan Keperawatan
Pada PASIEN
Dengan Gangguan
SISTEM IMUNOLOGI

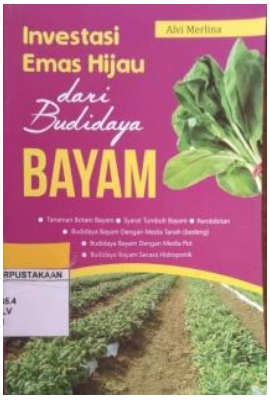

Dr. Ir. Kemas Ali Hanafiah, M.S.
Rancangan Percobaan
Teori & Aplikasi
Edisi Ketiga


Abdul Ghofar, S.Kep., Ns, M.Pd.
Pedoman Lengkap
KETERAMPILAN PERAWATAN
KLINIK
Untuk Dosen & Siswa/Mahasiswa Keperawatan

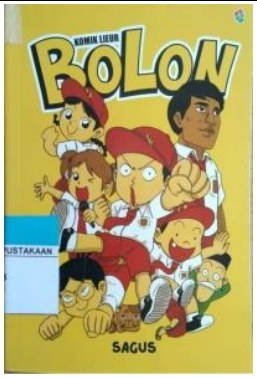

Feri Anwar
KIAT AMPUH
Bertanam
JAHE MERAH


ANDREW JOHAN
BUKU AJAR
BIOKIMIA
KEKEBALAN
TUBUH


Selalu
Ada
Harapan
Pengalaman Menangani
Anak Penderita Kanker

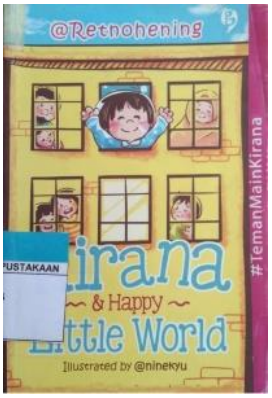

Investasi
Emas Hijau
dari
Budidaya
BAYAM
Alvi Merlina


Yahya Ahmad
RAHASIA
SUKSES
SEMANGKA
Berbisnis
dan Budidaya

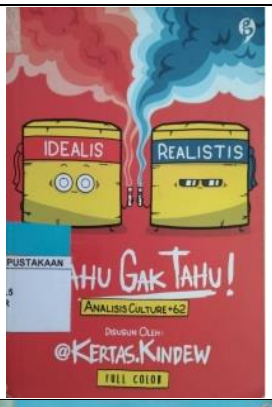

KOMIK LUCU
BOLON
SAGUS


GHOSTY's
KHILAFORIA
ヒラフホリア
GHOSTYCORP

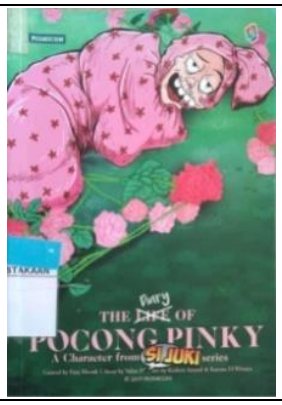

KAYA ITU
SUPER
MUDAH
Ragam Teknik Simpel Meraup
dan Menghemat Uang
Bukan Sekadar Teori!
Aditya Valentino

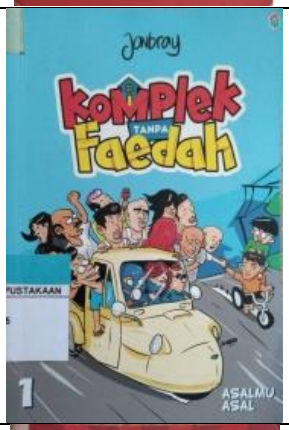

@Retnohening
#TemanMainKirana
irana
& Happy
ttle World
Illustrated by @nineKyu


PIONICON
Juleha
DAN GEROMBOLAN
GENG MOTOR


IDEALIS
REALISTIS
AHU GAK TAHU!
ANALISIS CULTURE +62
@KERTAS.KINDEW
FULL COLOR


EDISI SPESIAL
13
SI JUKI
KEROYOKAN

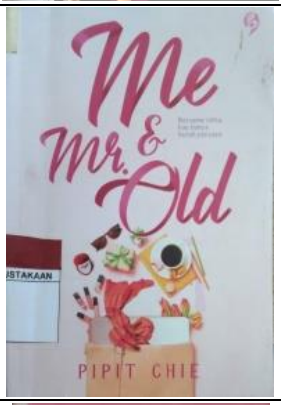

THE END OF
POCONG PINKY
A Character from SI JUKI series


Jonbray
komplek
TANPA
Faedah
1
ASALMU
ASAL


DIANDE FAISAL
MANJALI
4


Si nopal
and The Perpect
Family #2
oleh Naufal Faridurrazak


GHOSTY's
KHILAFORIA
ヒラフホリア
GHOSTYCORP

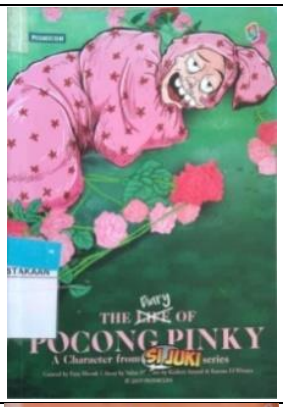

Me
&
Mr. Old
PIPIT CHIE

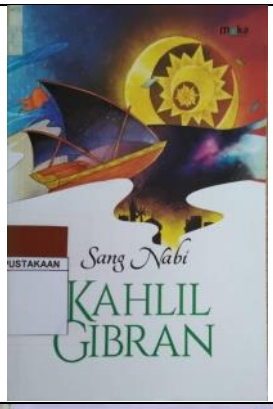

Amanda
Ami Rahmi

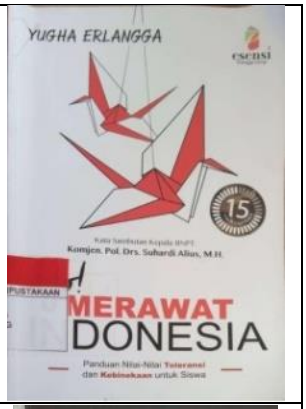

Juleha
DAN GEROMBOLAN
GENG MOTOR


rapuh
SEBUAH CINTA
PADA SENJA
KEDUA
Farah Nuruzzahra


TERE LIYE
Pergi

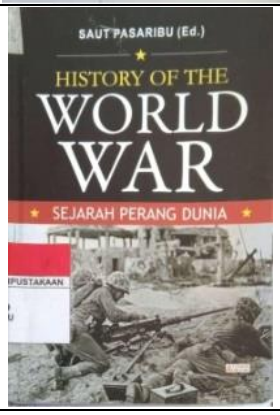

THE DIARY OF
POCONG PINKY
A Character from SIJUKI series

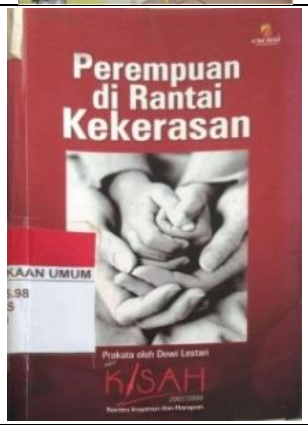

Sang Nabi
KAHLIL
GIBRAN


YUGHA ERLANGGA
MERAWAT
DONESIA

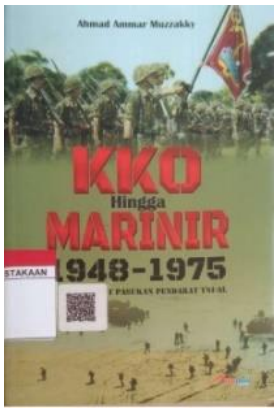

Si Nopal
and The Perfect
Family #2
oleh Naufal Faridurrazak

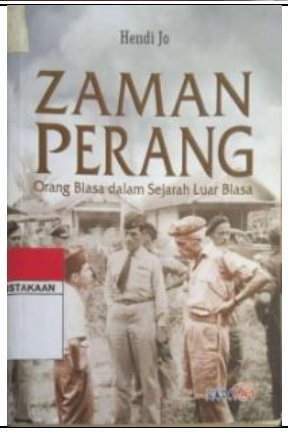

Iya, Pak!
Iya, Pak!
Iya, Pak!
Iya, Pak!
Iya, Pak!
Iya, Pak!
Iya, Sayang!
*IYA, PAK!
-HELYU L.A.

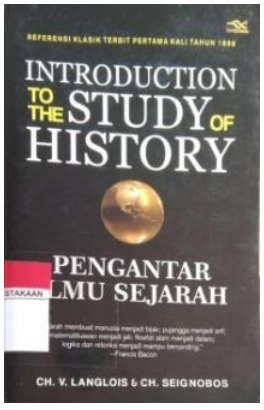

SAUT PASARIBU (Ed.)
HISTORY OF THE
WORLD
WAR
SEJARAH PERANG DUNIA

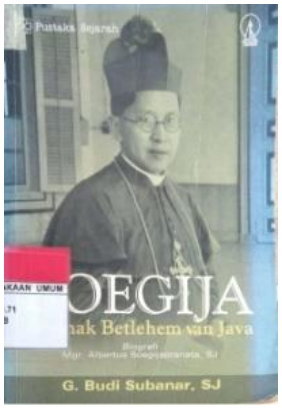

Perempuan
di Rantai
Kekerasan
Prakata oleh Dewi Lestari
KISAH

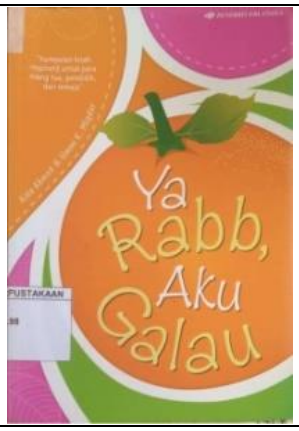

emir
KISAH DARI
BILIK
PESANTREN
15


Ahmad Ammar Muzzakky
KKO
Hingga
MARINIR
1948-1975


Hendi Jo
ZAMAN
PERANG
Orang Biasa dalam Sejarah Luar Biasa


INTRODUCTION
TO THE STUDY OF
HISTORY
PENGANTAR
LMU SEJARAH
CH. V. LANGLOIS & CH. SEIGNOBOS


OEGIJA
G. Budi Subanar, SJ

 Ya Rabb, Aku Galau PUSTAKAAN	 OSA KURNIAWAN ILHAM Proklamasi sebuah rekonstruksi	